

第十四届中国研究生数学建模竞赛

东华大学选拔赛

题 号 (A 题或 B 题) A 题

题 目 仓储物流中心选址与调度

团队成员 金元林 张东杰 刘如月

参赛队号 71

摘要

物流一直在生活中占据很重要的地位，货仓的选址以及配送的研究一直是物流行业的研究热点。人们总是希望能够提前获知物流量，安排好供货仓库和得到最佳配送路线以节约成本，实现利益最大化。

针对问题一，由于已知 A,B,C 三种商品在 20 个城市中三年的销售量，所以首先对已知的数据进行整理和规划以发现其规律如图 2.1。商品销售量的预测要考虑多方面的影响因子，比如淡旺季和长期趋势等。本文在数据归纳时发现三种商品的销售量都出现淡旺季和长期趋势，所以采用了基于时间序列分解的预测模型，将销售量影响因子分解出来，单独分析和预测，然后合并得到预测结果（见表 2.1 和附录一），比传统时间序列模型预测更加可靠。

针对问题二，根据预测结果获取满足供货的货仓总容量，然后结合货仓价格和供货约束条件建立数学模型。为解决全局规划的难题，仓储中心地址模型将系统聚类 and 重心法相结合，提出基于最短距离法和重心法的仓储中心选址流程。通过聚类分析方法对货仓供货区域进行划分（见图 3.5），约束供货区间，然后采用重心分析法对已划分的供货区域进行求解，寻找最优建仓位置。有效地解决了多节点仓储中心的选址问题。将“在一个区域内建多个仓储中心的选址问题”转化为“在多个区域内分别进行单一仓储中心选址问题”，将求解模型简化，减少了大量计算。最后，结合 A,B,C 商品的预测需求和建造及运营成本确定最终的城市和最佳仓储容量配比。

针对问题三，由于问题二中已经确定出仓储中心的最终位置和辐射范围，而且求解模型假设区域间供货互不交叉，所以首先根据需求量计算每个区域总供货车次，结合货车的约束条件建立模型。本文所建立的配送模型构成经典 TSP 问题，可以通过贪婪算法求解最短路径。然后利用约束条件下最优化方法，配置送货车次，得到最节省的送货方案。

最后，本文对三个问题的求解都是基于现有的优秀理论成果，而且对经典的算法进行过略微的调整和改进以适应本文中的模型需要。对问题二中复杂的最优求解问题，采用从局部到整体的方案，避免了算法复杂度过高的问题，但是同时可能造成模型求得的结果不是全局最优解。

关键词：时间序列分解，回归预测，系统聚类，重心法，最短路径

目录

摘要.....	1
1 问题重述.....	1
1.1 问题背景	1
1.2 问题的提出	1
2 问题一模型建立与求解.....	3
2.1 问题分析	3
2.2 商业销售预测的方法	7
2.2.1 时间序列分解预测模型的提出	7
2.2.2 时间序列分解预测模型的建立	7
2.2.3 销售数据变动的因素	7
2.2.4 时间序列模型的类型	8
2.2.5 时间序列分解预测模型的计算方法	8
2.3 销售量的预测	10
2.3.1 回归预测	10
2.3.2 残差分析	11
2.3.3 检验回归预测模型	13
2.3 总结	14
3 问题二仓储中心选址模型.....	15
3.1 问题描述	15
3.2 求解思路	15
3.3 仓储中心选址模型的建立与求解.....	16
3.3.1 二维地图坐标系的构建与区域的划分	16
3.4 仓储中心个数最优数学模型	21
3.5 重心法仓储中心选址模型	23
3.6 仓储中心选址模型小结	27
4 问题三车辆调度模型.....	28
4.1 问题三重述	28
4.2 问题分析	28
4.3 建立模型	30

4.4 模型求解.....	31
4.4.1 启发式算法	31
4.4.2 最短路径算法	31
4.4.3 TSP 问题求解最短路径	32
4.5 算例.....	34
4.5.1 算例描述	34
4.5.2 算例求解	35
4.6 结论.....	38
5 总结与展望.....	39
参考文献.....	40
附录.....	41

1 问题重述

1.1 问题背景

随着经济的发展，物流在未来的市场竞争中将起着举足轻重的作用，甚至被有些学者成为“第三利益源泉”。

与发达国家相比,我国的物流产业效率较低。在货物集装、运输、配送等方面不仅存在很多的费用消耗而且时间消耗巨大，这些费用和时间上的消耗和大量存在的“库存”正是潜在的实施物流管理的领域,这为物流的发展留下了巨大的空间。在这种形势下,研究如何通过实施科学的物流管理,以提高物流效率、降低物流成本、提高服务质量是十分必要的。

在物流配送业务中，仓储中心选址和车辆路径优化是关键环节，它对提升企业服务质量，降低物流成本，增加经济效益有重要作用，但涉及面广，需考虑的因素也较多。因此，如何根据实际情况求解最优选址和车辆最优路径，并建立车辆优化调度系统成为企业和学者的重要研究方向。合理的物流配送管理系统应满足如下三个目标：

- (1) 标准时运输。即货物按客户要求的指定时间段运到客户手中。
- (2) 车辆运输总路径最短。
- (3) 总成本（包括运输路线成本, 建立和运营仓库的固定成本, 获得运输工具成本等）最小。

1.2 问题的提出

M 公司在全国有 20 座城市拥有门店，销售 A, B, C 三种产品。现只有一个物流仓储中心进行负责全国配货。由于业务扩张和成本控制，现公司考虑在全国选取某些城市设立几个仓储中心，可以有效协调门店各产品的供需平衡，同时控制仓储物流成本。

问题 1：根据附件信息中 2014-2016 年 20 个城市 A, B, C 三种商品每月的销售量来预测 2017 年三种商品 12 个月每个月的销售数据。

问题 2：为提高整体供货效率，M 公司综合城市交通和地价因素，计划从 31 座城市选址筹建新的仓储中心。

公司对仓储中心有以下要求：

- (1) 仓储中心距离最远的城市不能超过 1000 公里，以满足城市突发的调货需求。
- (2) 所有仓储中心的总容量必须满足每月全国的需求总量。
- (3) 一个城市只能建造一个仓储中心，但可从多个中心尽心配货。
- (4) 每个仓储中心的一次性建设成本为 150 万元，可分 10 年分摊，每月的维护成本如下表 1.1。

表 1.1 容量与维修成本对应表

容量	维护成本/月
0-1000 单位	5000 元
1000-3000 单位	10000 元
3000-5000 单位	18000 元
5000 单位以上	每增加 1000 单位，增加 2500 元

由第一问预测出的需求量，给出仓储中心的位置，数量，规模，以及每个仓储中心三种产品的容量配比，已达到成本最优。

问题三：每个仓储中心建成后都需要一定数量的货车完成每月的运货任务，已知：

- (1) 货车的采购成本为 15 万元，报废周期为 6 年，每月维护费用为 1000 元，运输费用为 150 元/百公里。
- (2) 每辆货车可装载的最大容量为 250 单位产品。
- (3) 每辆货车每天可行驶 500 公里。
- (4) 保证每座城市在 7 天能获得一轮配货。

请在第 2 问确立的仓储中心的基础上，运用合理假设，建立模型，给出每个仓储中心需要配置的货车数和行驶路线，在成本最优的条件下满足常规的送货需求。

2 问题一模型建立与求解

2.1 问题分析

问题一，主要是根据 2014-2016 年 20 个城市 A, B, C 三种商品每月的销售量来预测 2017 年三种商品 12 个月每个月的销售数据。通常，在预测商品销售的时候首先要考虑这三种商品的销售受哪些因素的影响，比如服装和水果的销售一般会受到地域和季节的影响，所以首先要确定商品是否是季节性产品。其次，同类商品的销售一般相互影响，所以要考虑三种商品之间的销售是否相互影响，20 个城市之间的销售是否相互影响。然后，要根据数据分析商品的变化趋势和变化规律，分析在预测时如何综合考虑这种变化的规律。最后，要考虑的如何选择预测模型，预测方法的研究成果比较多，适用的情况也不相同，所以要首先提取数据，分析数据的分布特点和变化规律选择合适的预测模型。

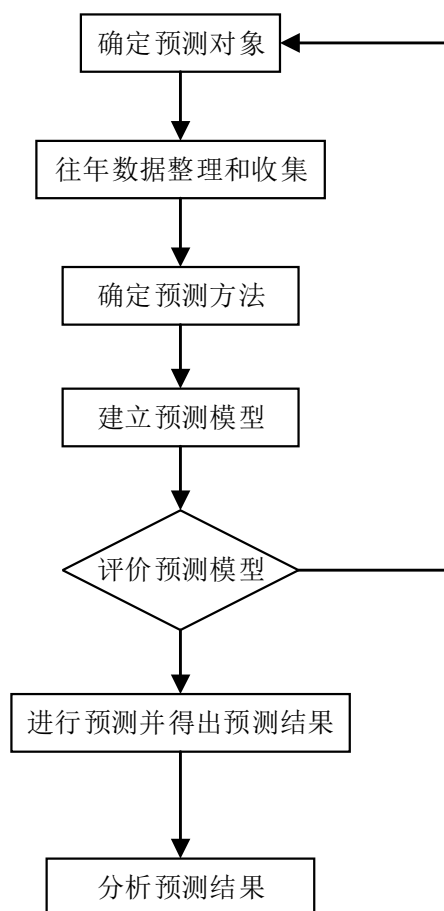


图 2.1 模型建立流程

本文的预测对象是 A,B,C 三种商品在 2017 年 12 个月所有城市的销售量，每个城市都拥有这种三种商品。需要整理和分析的数据是 2014 到 2016 年这三年的销售数据，主要是将这些数据重新整理归纳一下，以发现其内在的规律。预测方法的选择和预测模型的确定是本节最关键的内容，因为只有选择合适的预测模型才能够得到比较合理的预测数据。预测模型的评价主要是通过比较 2014 年到 2016 年的实际值和预测值之间的差值，预测模型选择的越好，那么误差也会越低，所以再保证模型合理的情况尽量选择误差较小的预测方法。

历史数据的获取和观测：

通过使用 Matlab 可以对数据做初步的统计和可视化展示，为分析预测模型做准备，如下图所示的数据是为了展现其长期的变化规律。

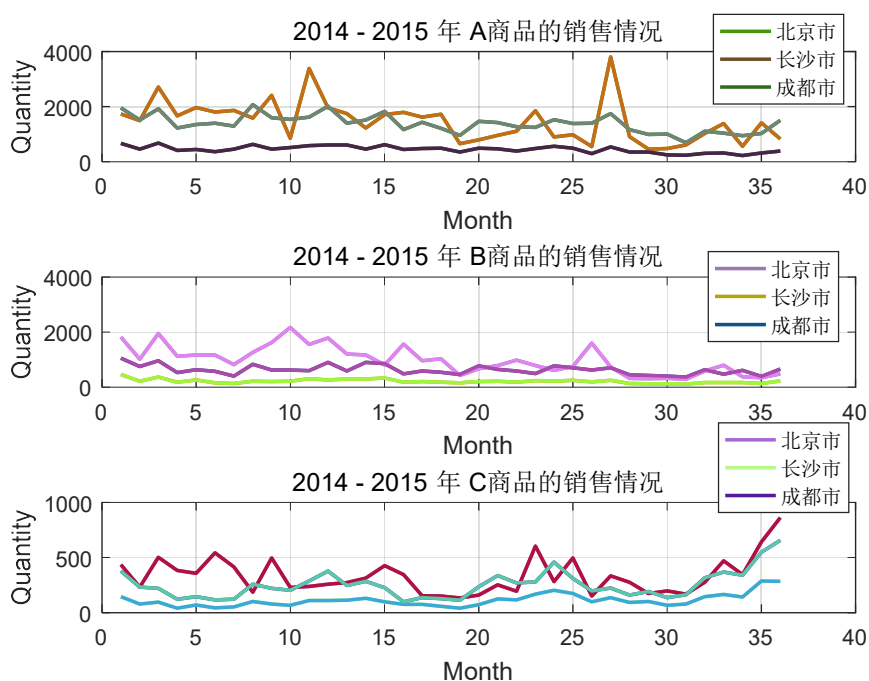


图 2.2 A,B,C 三种商品的三年销售状况

由于数据比较多，有二十个城市的数据没办法全部展示，从中随机选择了三个代表城市，北京、长沙、成都市的 2014-2015 年的销售数据，从图中可以明确的看出，三年的销售量是有很大的波动，而且有比较明显的变化趋势。比如 A 商品，变化的幅度比较大，而且可以明显的看出，其三年的销售量在逐渐的减少。B 商品的三年销售量同样也是可以看出在逐步的减少，而 C 商品的最终趋势可以看出，在未来一年里，销售量很可能会有所增加。所以从变化的趋势图可以推测出，A,B,C 三种商品的有个长期趋势的，在预测的时候首先要考虑其长期的变化趋势。

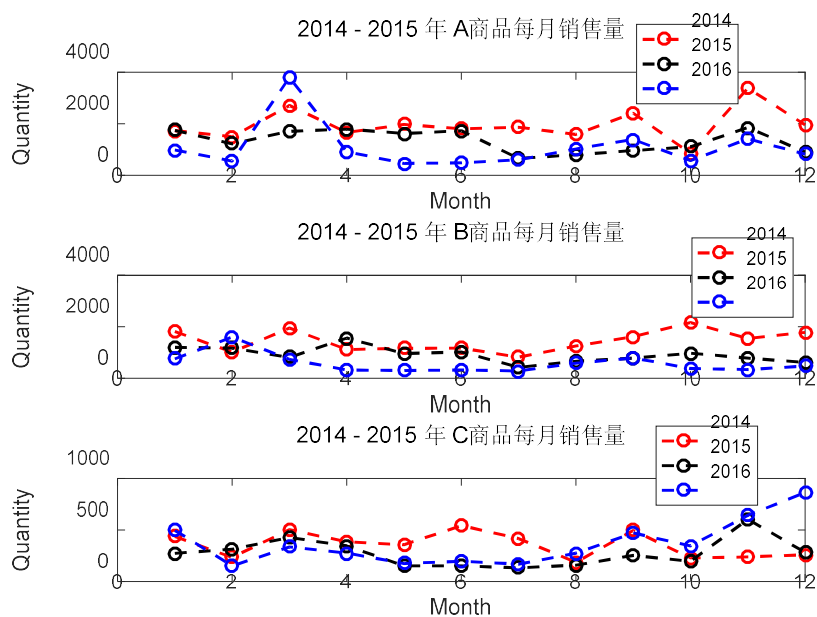


图 2.3 三种商品三年中每个月份的销售情况

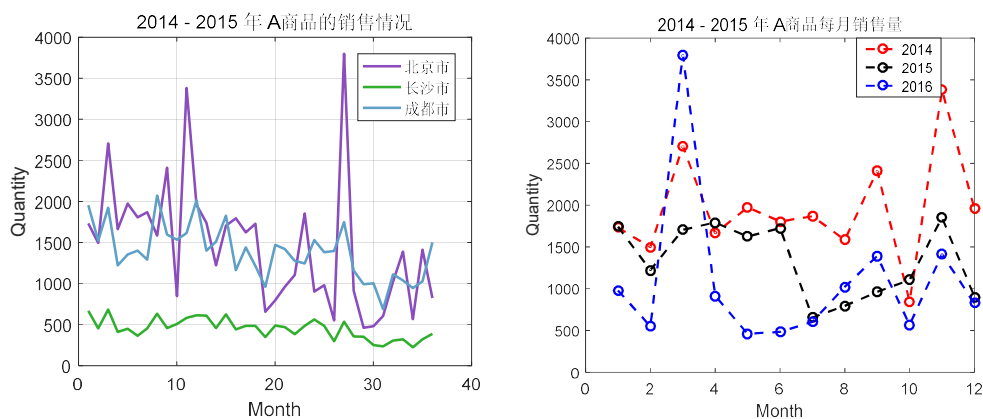


图 2.4 A 商品三年里的变化状况对比

图 2.3 和图 2.4 是为了展示三种商品从 2014 年到 2016 年的 1 月到 12 月的变化情况，主要是为了突出商品的销售量在个别月份的一个比较大规模的波动。从图 2.4 可以明显的看到，在 3 月份、9 月份、11 月份 A 商品在这三年都有一个比较剧烈的波动，这样可以间接的说明，A 商品的销售明显的受到了月份的影响，或者也可以说是受到了季节的影响。虽然 B 商品的此类变化不是很明显，但是 C 商品也可以明显看出是受到了月份的影响。因此我们可以任务，A,B,C 商品的销售会受到月份的影响，也就是说有淡季和旺季。和长期趋势一样，淡季和旺季对商品的预测至关重要。

从 2014 年到 2016 年 A,B,C 类商品的总体的销售变化趋势有比较下降或上

升，三种商品而且在个别的月份也有明显的上升和下降，这说明月份或季节性的因素同样比较明显，所以在设计预测模型是要充分考虑趋势和月份因素影响（淡季和旺季）。

2.2 商业销售预测的方法

2.2.1 时间序列分解预测模型的提出

国内外对销售预测的研究有很多，比较主流的预测方法有，神经网络法，回归模型，移动平均法，指数平滑方法，包括最近热门的深度学习等都可以根据往期的数据对销售进行预测。本文的模型是建立在时间序列模型的基础上，根据时间序列的特点，找到这些序列的发展趋势和内在特性，再提取出准确的信息，比较经典的时间序列技术有滑动平均模型和指数平滑模型等。

2.2.2 时间序列分解预测模型的建立

模型假设：

- （1）假设同种商品在 20 个城市的需求的变化规律相同，即具有相同的销售模型
- （2）假设 A,B,C 三种商品在同一个城市的销售符合一个模型
- （3）假设 A,B,C 三种产品的销售相互不影响，20 个城市销售彼此间也不相互影响

社会经济现象是不断发展变化的，经济时间序列的变化受到长期趋势 T、季节变动 S、周期变动 C 和不规则变动 I 这四个因子的影响，表现为数量上的波动。

时间序列预测技术就是通过对预测目标自身时间序列的处理，来研究其变化趋势

的。一个时间序列往往是以下几类变化形式的叠加或耦合，函数表达式为：

$$y_t = f(T_t, S_t, C_t, I_t) \quad (2-1)$$

2.2.3 销售数据变动的因素

- （1）长期趋势变动

长期趋势因素反映的是事物的主要变化趋势，表现为在一段时间范围内持续不断地向某方面发展。

(2) 季节变动

指的是淡旺季月份，或者受季节的影响。

(3) 循环变动

呈周期性的变化趋势。

(4) 不规则变动

受众多偶然因素的影响而出现的随机变动。

2.2.4 时间序列模型的类型

比较常用的时间序列模型如要有以下两种：

(1) 加法模型

$$y_t = T_t + S_t + C_t + I_t \quad (2-2)$$

(2) 乘法模型

$$y_t = T_t \times S_t \times C_t \times I_t \quad (2-3)$$

2.2.5 时间序列分解预测模型的计算方法

(1) 计算季节指数 S 的值

计算季节指数 S ，首先利用移动平均法将长期趋势 T 和周期变动 C 两大变动因素剔除掉然后再运用按季平均法即可求出季节指数 S 。

将各年的季度销售数据按照时间先后顺序依次排列开来，计算 12 个月份的移动平均。移动平均包括普通移动平均和加权移动平均。

将各年相对应的月份的销售数据 y_t 除以序列 TC 得到序列 SI ，然后把 SI 重新按照月份进行排列，运用移动平均法求出月份的平均数，再加以修正，即可以得到各个季节的指数 S_1, S_2, S_3, S_4 。

1. 简单移动平均法：

设观测数据的序列为 y_1, \dots, y_T ，取移动平均的项数 $N < T$ ，一次简单的移动平均值的计算方是：

$$\begin{aligned}
M_t^{(1)} &= \frac{1}{N}(y_t + y_{t-1} + \dots + y_{t-N+1}) \\
&= \frac{1}{N}(y_t + y_{t-1} + \dots + y_{t-N}) + \frac{1}{N}(y_t - y_{t-N}) = M_{t-1}^{(1)} + \frac{1}{N}(y_t - y_{t-N})
\end{aligned} \quad (2-4)$$

2. 加权移动平均计算

在简单移动平均公式中，每期数据在求平均时的作用是等同的。但是，每期数据所包含的信息量不一样，近期数据包含着更多关于未来情况的信息。因此，把各期数据等同看待是不尽合理的，应考虑各期数据的重要性，对近期数据给予较大的权重，这就是加权移动平均法的基本思想。

设时间序列为 $y_1, y_2, \dots, y_t, \dots$ ；加权移动平均的公式为：

$$M_{tw} = \frac{w_1 y_t + w_2 y_{t-1} + \dots + w_N y_{t-N+1}}{w_1 + w_2 + \dots + w_N}, t \geq N \quad (2-5)$$

其中 M_{tw} 为 t 期加权移动平均数； w_i 为 y_{t-i+1} 的权数，权值的大小决定了 y_i 所占据的比重，也就是其在整个移动周期的重要性。

(2) 计算长期趋势 T 的值

长期趋势的值可以通过回归预测来获取，回归方程的选择有很多，例如，多项式回归，直线回归以及指数回归等，具体的选择主要是根据数据的特性。回归方程的决定了数据预测的准确性，本文根据数据的变化规律选择的回归方式是多项式方法。

(3) 计算周期变动 C 的值

以序列 TC 除以长期趋势 T 即可得到周期变动因素 C 的值

(4) 计算不规则变动 I 的值

不规则 I 的值一般很难直接确定，所以主要是通过分离出来长期趋势 T，季节指数 S 和周期变动 C 后通过公式 $I = Y / TSC$ 计算得到。

2.3 销售量的预测

2.3.1 回归预测

本文在进行销售预测的过程中，没有采用完全分离四种变化因子的方法，而是通过首先分离季节因子 S ，然后获得 TCI 的值，通过预测 TCI 的值再乘以相应的季节指数 S 的值，就可以得到最后的预测结果。常用的回归模型是线性回归，线性模型主要是用来预测数据的分布呈线性的情况，但是本文中的离散点图可以看出，线性的状态并不明显，所以采用非线性的预测方法。

(1) 明确预测的具体目标

如预测具体目标是下一年度的销售量（ TCI ，并不是真正的销售量），那么销售量 TCI 就是因变量根据预测目标，确定自变量和因变量。因为销售量与 TCI 因子的关系是：

$$Y = TCIS \quad (2-6)$$

(2) 建立回归预测模型

依据自变量和因变量的历史统计资料进行计算，在此基础上建立回归分析方程，即回归分析预测模型。回归预测模型也就是确定回归方程，最常用的回归方程式线性回归方程，主要的优点是比较简单。但是从过滤月份因子得到的散点图可以看出，简单的线性回归并不能很好的进行拟合，所以本文中采用非线性的拟合方法。

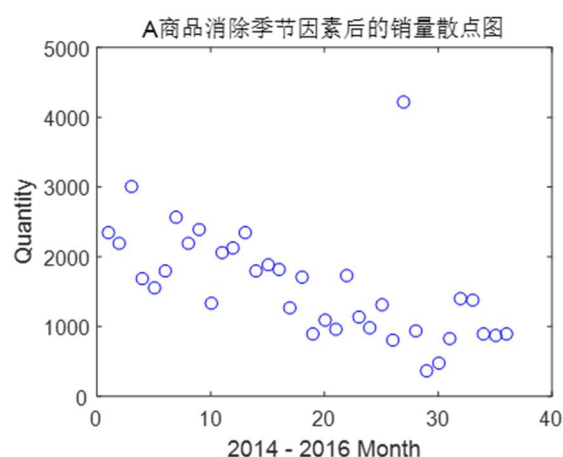


图 2.5 A 商品消除 S 后的散点图

1. 多项式回归模型

多项式回归主要是用来研究一个因变量与一个或多个自变量间多项式的回

归分析方法，如果自变量只有一个时，称为一元多项式回归。本文的自变量是时间（月份），也就是只有一个自变量，所以本文所使用的是一元多项式的回归。

$$\hat{y} = b_0 + b_1x + b_1x^2 + \dots + b_mx^m \quad (2-8)$$

多项式回归虽然通常可以达到比较好的预测效果，但是如果拟合的过好，容易出现过拟合的问题，所以多项式的次数不能过高，而且要根据实际的数据特征选择适合的回归方程。所以本文中根据实际的数据分析，选择的拟合曲线是二次拟合。本文拟合的主要目的只是为了获得一个变化趋势，也就是 *TCI*，所以没有必要做采用高次曲线进行拟合。其表达式可以表示为：

$$\hat{TCI} = b_0 + b_1x + b_1x^2 \quad (2-9)$$

$$\hat{y} = \hat{TCI} \quad (2-10)$$

2.3.2 残差分析

残差分析的目的主要是通过残差所提供的信息，分析出数据的可靠性、周期性或其它干扰。用于分析模型的假定正确与否的方法。所谓残差是指观测值与预测值（拟合值）之间的差，即是实际观察值与回归估计值的差。

本文在第一次拟合做残差分析主要是为了消除误差比较大的点。误差点对预测曲线的影响比较大，所以为了能够获得比较好的拟合效果要对其做残差分析。如下图所示的就是残差分析得到的结果，可以看出红色的部分就是需要消除的误差点。



图 2.6 A 商品残差分析

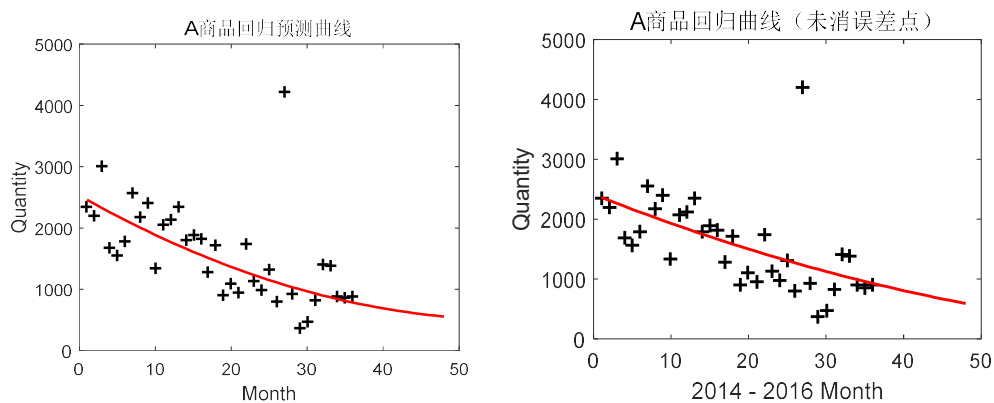


图 2.7 A 商品消除误差点前后的拟合曲线对比

图中可以看到，消除误差点后拟合曲线更靠下方一些，下降的趋势更加平缓，所以误差点的影响对长期的变化趋势的影响很大。而且消除误差点的影响，也能得到更加准确的预测数据。所以每次回归预测时要进行残差分析，消除误差点的影响。

（3）进行相关分析

回归分析是对具有因果关系的影响因素（自变量）和预测对象（因变量）所进行的数理统计分析处理。只有当变量与因变量确实存在某种关系时，建立的回归方程才有意义。因此，作为自变量的因素与作为因变量的预测对象是否有关，相关程度如何，以及判断这种相关程度的把握性多大，就成为进行回归分析必须要解决的问题。进行相关分析，一般要求出相关关系，以相关系数的大小来判断自变量和因变量的相关的程度。

2.3.3 检验回归预测模型

回归预测模型是否可用于实际预测，取决于对回归预测模型的检验和对预测误差的计算，所以要在数据预测时首先对预测误差进行计算。利用回归预测模型计算预测值，并对预测值进行综合分析，确定最后的预测值。

常用的回归误差的计算方法有以下几种：

1. 平均误差

$$\bar{e} = \frac{1}{n} \sum_{i=1}^n e_i \quad (2-11)$$

2. 平均绝对误差 MSE

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (2-12)$$

3. 标准误差 RMSE

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (2-13)$$

因为本文预测的结果并不是直接的销售量，而是长期趋势指数 \hat{TCI} 值，所以本文在求回归误差的时候并不需要计算所有的误差，为了简化计算量，本文采用的误差表示方式是标准误差，所以可以根据公式得到 TCI 的标准差误差，标准差的计算结果由于过多无法全部列出，请见附录二。预测及计算结果如下所示：

表 2.1 十个城市的销售需求量预测结果

月份	前十个城市的 12 个月的需求量预测结果									
城市	北京	长沙	成都	广州	贵阳	哈尔滨	杭州	昆明	南京	宁波
1	665	225	797	351	162	111	579	250	686	88
2	589	213	1005	364	214	138	444	206	401	129
3	754	228	1078	404	235	155	465	216	385	116
4	796	160	801	414	139	132	348	158	529	104
5	988	172	867	391	195	133	702	118	671	153
6	755	169	888	418	131	102	413	191	467	123
7	525	101	637	340	117	82	238	152	300	64
8	504	120	981	342	135	137	438	148	497	97
9	672	81	821	380	170	120	214	161	328	100
10	408	59	741	336	103	110	243	118	463	67
11	1012	47	733	401	158	122	478	106	482	116
12	544	24	886	411	96	122	207	169	452	91

由于商品的数据较多，所以这里只给出 A 类商品的前十个城市的预测结果，其他的预测结果请见附录一。

2.3 总结

预测结果以及模型的评价标准主要有两个方面，第一个是模型合理性的分析和检验，主要包括以下两点：第一，预测结果的长期趋势要和以上三年的整体趋势一致；第二，预测的周期性变化和季节性变化要符合商品的规律。第二个评价标准是检查预测的精度，销售预测的精度，指的是销售对象的预测值与实际值之间的误差或偏差程度，误差越大，说明预测精度越低，误差越小，说明预测精度越高，预测结果越好。对销售预测精度的测定就是对销售预测误差的测定，常用来测定销售误差的方法有在以上已经有过介绍。所以本文采用的预测模型考虑了长期趋势和淡旺季的需求，对模型进行了优化，从后面得到的结果可以看出预测的效果比较理想。

3 问题二仓储中心选址模型

在问题 1 中，我们通过 MATLAB 软件算法完成了 2017 年度每月 A、B、C 三种产品在 20 座城市销量的预测。在预测出 20 座城市每月 A、B、C 三种产品的需求量之后，要根据需求量和相关约束条件对仓储中心的最优地址及其数量进行选取。

3.1 问题描述

为提高整体供货效率，M 公司综合城市交通和地价因素，计划从以下城市选址筹建新的仓储中心。

公司对仓储中心有以下要求：

- (1) 仓储中心距离最远的城市不能超过 1000 公里，以满足城市突发的调货需求。
- (2) 所有仓储中心的总容量必须满足每月全国的需求总量。
- (3) 一个城市只能建造一个仓储中心，但可从多个中心尽心配货。
- (4) 每个仓储中心的一次性建设成本为 150 万元，可分 10 年分摊，每月的维护成本如下表 3.1。

表 3.1 容量与维护成本对应表

容量	维护成本/月
0-1000 单位	5000 元
1000-3000 单位	10000 元
3000-5000 单位	18000 元
5000 单位以上	每增加 1000 单位，增加 2500 元

由第一问预测出的需求量，给出仓储中心的位置，数量，规模，以及每个仓储中心三种产品的容量配比，已达到成本最优。

3.2 求解思路

该问题属于典型的多源选址问题，即已知若干个现有备选仓储中心选择多个仓储中心最优地址问题。多源选址问题的一般方法是：根据已知的各个用户终点的位置与需求量以及区域内每条路径的运费，求解出源的个数、各个源的位置及

其容量的问题。

由于该问题 2 暂时并未涉及产品从仓储中心到各销售门店所需的运输费用，但是问题 3 要求解出货车的数量及其路线，考虑到为节约货车运输成本，本模型采用用系统聚类法和重心法选址模型相结合的方法，完成多节点配送中心的选址问题。仓储中心选址模型分为以下几个步骤：

第一步：根据 20 个销售门店的地理位置以昆明为坐标原点进行了二维地图坐标的构建，并且对各个销售门店依次进行编号。然后利用系统聚类法中的最短距离法完成对销售门户的分类，将全国 20 个销售门户所在城市分成不同的子区域，将“在一个区域内建多个仓储中心的选址问题”转化为“在多个区域内分别进行单一仓储中心选址问题”，将求解模型简化，减少了大量计算。在建立好系统聚类法模型后，使用 SPSS 软件中系统聚类分析法中的最短距离法进行模型的求解与仿真，在满足约束条件的前提下，对所有销售门店进行区域划分、对模型求解。

第二步：建立仓储中心个数最优数学模型，求解出在不同销售需求范围内，如何建造仓储中心使其成本最优。

第三步：建立重心法仓储中心选址模型，利用重心法选址模型确定各子区域配送中心的具体位置。即将划分好的区域中各个销售门店的需求量作为模型的输入信息，通过重心模型求解出各个区域中的最优解，即各个区域仓储中心的坐标。由题目假设知不考虑城市的实际尺度，使城市简化为点处理，若求解出的仓储中心坐标位置偏离备选位置，可首选临近销售需求量较大的门户，以此减少该区域的运输费用。

3.3 仓储中心选址模型的建立与求解

3.3.1 二维地图坐标系的构建与区域的划分

二维地图坐标系的构建：

由于本数学模型会涉及到区域的划分以及第三个问题中运输路线和费用的求解，因此首先对 20 个销售门户在中国地图上标出并进行依次编号，如下图 3.2 所示。



图 3.2 各销售门户坐标图

然后以昆明为坐标原点，以正东方为 x 正半轴，正北方为 y 正半轴，以 1 公里为单位，建立二维坐标系。其中各个销售门户城市的 (x, y) 坐标可根据中国地图依次测得数据。在测量过程中可能会存在测量误差，在此忽略不计。各销售门户坐标与建立的二维坐标系如下表 3.3 所示。

表 3.3 各销售门户精确坐标表

城市	x/km	y/km
北京	1382.29	1640.61
长沙	1024.17	363.17
成都	115.46	644.68
广州	1068.58	196.59
贵阳	403.71	180.74
哈尔滨	2423.6	2292.01
杭州	1709.1	624.04
昆明	0	0
南京	1620.33	767.44
宁波	1849.46	576.4
上海	1867.3	663.36
深圳	1156	-269
沈阳	2024.06	1881.35
石家庄	1144.19	1471.28
天津	1465.23	1555.02
温州	1758.77	368.75
武汉	1173.99	613.92
西安	616.09	1018.55
郑州	1091.13	1097.18
重庆	390.28	517.17

把上述测得各城市坐标点转换到二维坐标系中，结果如下图所示。

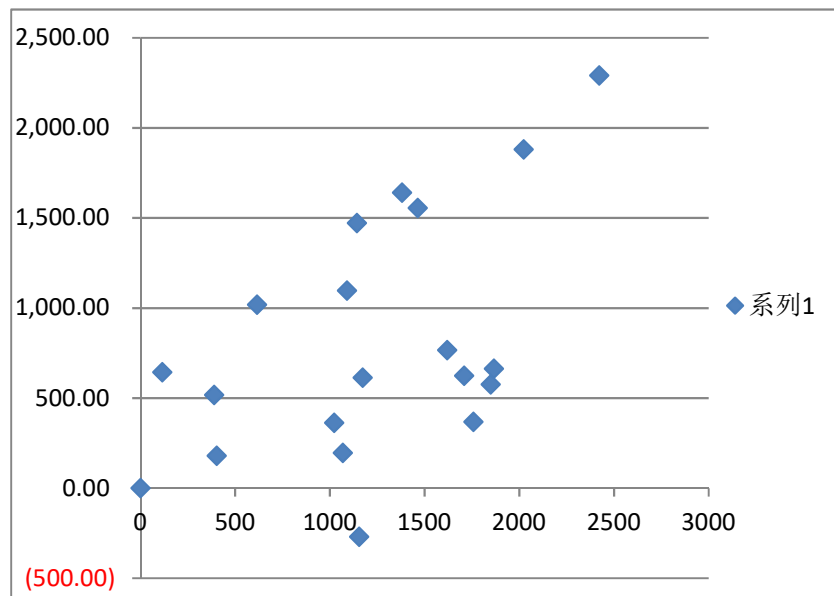


图 3.4 各销售门户二维直角坐标图

区域的划分:

由于本模型中的销售门户较多且分布广, 因此先对全国所有的销售门户进行区域划分, 形成若干配送区域, 将一个多配送中心选址问题转化为多个单配送中心选址问题, 简化需要求解的模型。

仓储中心的选址影响因素有很多, 如物流费用、商品特性、自然条件等因素。本文选取主要的宏观因素, 根据销售门户之间的宏观因素的相似性将销售门户分类, 使相似性较大的销售门户划分在同一区域, 即根据各个销售门户的坐标信息对其进行分类。分类的方法采用的是系统聚类法中的最短距离法。

聚类分析的基本原理为: 聚类分析是根据事务本身的特征研究个体分类的基本方法, 其基本原则就是同一类中的个体有较大的相似性, 而不同类的相似性较差或者较大。聚类方法主要有系统聚类法、有序样品聚类法、动态聚类法、模糊聚类法、图论聚类法、聚类预报法等。本文运用系统聚类中的最短距离法, 其基本步骤和原理表述如下:

- (1) 以绝对值距离作为样本间的距离, 以最短距离作为类间的距离;
- (2) 计算 N 个样本两两之间的距离, 得到距离矩阵;
- (3) 构造个类, 每类只含有一个样本;
- (4) 合并符合类间距离定义要求的两类为一个新类;
- (5) 计算新类与当前各类的距离。若类的个数为 1。则转到步骤 6, 否则回到步骤 4;
- (6) 画出聚类图;
- (7) 决定类的个数和类。

绝对值距离和类间的最短距离分别表示如下:

$$d(x, y) = \sum_{j=1}^p |x_j - y_j| \quad (3-1)$$

$$D(G_1, G_2) = \min_{\substack{x_i \in G_1 \\ y_j \in G_2}} \{d(x_i, y_j)\} \quad (3-2)$$

为对 20 个销售门户根据位置信息进行分类, 因此将建立好的各销售门户坐标数据信息输入 SPSS 软件, 在输入好信息后在分析界面中找到系统聚类分析, 在方法中选择最短距离法, 点击保存, 进行模型求解, 求解运算后的聚类谱系图

如图 3.5。

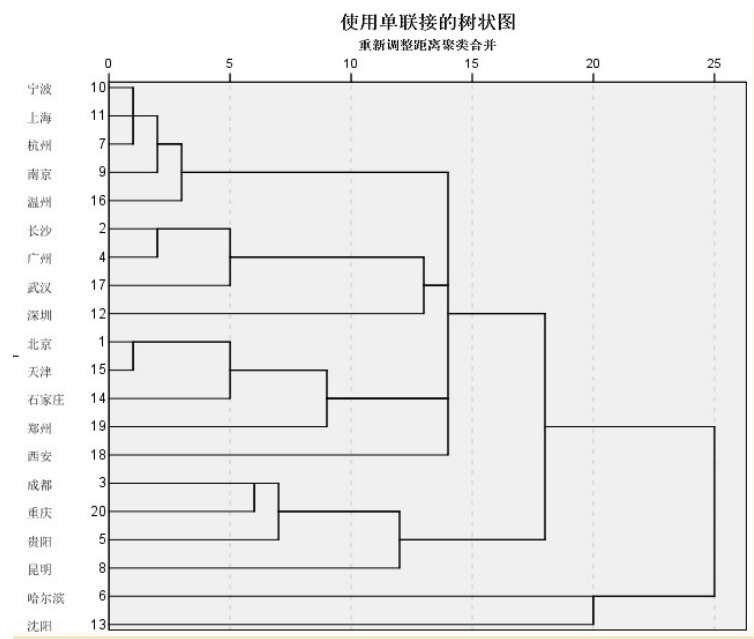


图 3.5 系统聚类分析图

由约束条件知，仓储中心距离最远的城市不能超过 1000 公里，以满足城市突发的调货需求。因此由系统聚类谱系图可根据每个门店之间距离，将 20 个销售门户划分为五个区域，能更好的对各类区域具体分析并规划对应的配送中心，选择合理的配送方式，各销售门店区域划分与 2017 年各门店每月需求量与各地区 2017 年每月需求总量如下表 3.6。

表 3.6 各区域需求量表

区域	序号	地区	x 坐标	y 坐标	需求量	总需求量
一	6	哈尔滨	2424	2292	5357	9603
	13	沈阳	2024	1881	4246	
二	3	成都	115	645	22523	45155
	5	贵阳	404	181	4260	
	8	昆明	0	0	5208	
	20	重庆	390	517	13164	
三	1	北京	1382	1641	19763	47754
	14	石家庄	1144	1471	2893	
	15	天津	1465	1555	10225	
	18	西安	616	1019	8792	
	19	郑州	1091	1097	6081	
四	2	长沙	1024	363	6364	26905
	4	广州	1069	197	8984	
	12	深圳	1156	-269	3454	
	17	武汉	1174	614	8103	
五	7	杭州	1709	624	16764	57523
	9	南京	1620	767	11567	
	10	宁波	1849	576	2892	
	11	上海	1867	663	21229	
	16	温州	1759	369	5071	

3.4 仓储中心个数最优数学模型

(1) 模型相关参数一览表如下

表 3.7 模型相关参数表

序号	符号	意义
1	m	仓储中心个数
2	n	销售门店个数
3	b_{jh}	第 j 个销售门店对 h 产品的需求量
4	a_i	第 i 个仓储中心的总容量
5	h	产品种类
6	e_{a_i}	仓储中心不同容量 a_i 下的维修成本
7	$G(x)$	满足销售总量时建造仓储中心总费用

(2) 目标函数

在该模型中，总费用 $G(x)$ 由以下两部分组成：第一部分为仓储中心一次性建设成本，已知仓储中心一次性建设成本为 150 万元，可分 10 年分摊，因此可求得一个仓储中心每月需还的建造成本为 1.25 万元，则 m 个仓储中心每月需还的建造成本为 $1.25m$ 万元。第二部分为 维修成本费用，仓储中心不同容量 v 下的维修成本也不同。维护成本与容量对应公式如下：

$$a_i = \begin{cases} 0.5 \text{万元}, & 0 < a_i \leq 1000 \\ 1 \text{万元}, & 1000 < a_i \leq 3000 \\ 1.8 \text{万元}, & 3000 < a_i \leq 5000 \\ \frac{a_i - 5000}{1000} \times 0.25 \text{万元}, & 5000 < \frac{a_i}{m} \end{cases} \quad (3-3)$$

则总费用 $G(x)$ 为上述两费用之和，根据成本最优原则，使总费用 $G(x)$ 最小，即

$$\text{Min}G(x) = \min \left[\sum_{i=1}^m (1.25 + e_{a_i}) m \right] \quad (3-4)$$

(3) 约束条件

1. 需求约束

在运送产品的过程中，各个区域中仓储中心向区域内销售门店的运送各种类货物的总量应满足区域内销售门店的总需求，即：

$$\sum_{i=1}^m e_{a_i} \geq \sum_{j=1}^n \sum_{h=1}^3 b_{jh} \quad (3-5)$$

2. 非负约束

在该模型中，涉及到的相关变量必须大于或等于零，即：

$$\sum_{i=1}^m e_{a_i} > 0, \sum_{j=1}^n \sum_{h=1}^3 b_{jh} > 0, a_i > 0, j = 1, 2, 3, \dots, n, h = 1, 2, 3, i = 1, 2, 3, \dots, m \quad (3-6)$$

3. 距离约束

仓储中心距离最远的城市不能超过 1000 公里，以满足城市突发的调货需求。

综上所述建立仓储中心个数最优数学模型为

$$MinG(x)=\min \left[\sum_{i=1}^m (1.25+e_{a_i}) m \right] \quad (3-7)$$

S.T.

$$\sum_{i=1}^m e_{a_i} \geq \sum_{j=1}^n \sum_{h=1}^3 b_{jh} \quad (3-8)$$

$$\sum_{i=1}^m e_{a_i} > 0, \sum_{j=1}^n \sum_{h=1}^3 b_{jh} > 0, a_i > 0, j = 1, 2, 3, \dots, n, h = 1, 2, 3, i = 1, 2, 3, \dots, m \quad (3-9)$$

(4) 模型求解结论

该模型为典型的不允许缺货模型，经求解得出，在需求量固定且不考虑运输路线与运输费用的前提下，建造的仓储中心个数越少成本价格越优。所以，可以得出结论，在划分好的区域中，每个区域中建造一个仓储中心时成本为最优。

3.5 重心法仓储中心选址模型

由上述知，已将 20 个销售门户划分为五大区域，经计算得出每个区域中建造一个仓储中心时成本为最优，则已经把“在一个区域内建多个仓储中心的选址问题”转化为“在多个区域内分别进行单一仓储中心选址问题”。单仓储中心选址方法中，虽然仓储中心选址方法很多，但利用数学方法对物流进行选址是较为准确的方法，重心法是常用的方法之一，适用于单中心选址问题^[5]。因此

(1) 重心法选址基本原理

重心法又称网格法或精确重心法，它能利用物理学中对一组离散质点求解重心的原理来解决设施选址的问题。在生产成本中运输费用占有较大比重的情况下，由一个工厂向多个配送中心或仓库发货，或由一个配送中心或仓库向多个销售点运货，都适宜于采用重心法。其目标函数为：

$$MinTC = \sum_{i=1}^n f_i V_i d_i \quad (3-10)$$

其中

$$d_i = \sqrt{(X_c - d_{ix})^2 + (Y_c - d_{iy})^2} \quad (3-11)$$

对 d_{ix} 和 d_{iy} 求偏导，令偏导数为零，即：

$$\frac{\partial TC}{\partial d_{ix}} = 0, \frac{\partial TC}{\partial d_{iy}} = 0 \quad (3-12)$$

由此可解得 X_C 和 Y_C 的解为：

$$X_C^* = \frac{(\sum_{i=1}^n f_i V_i d_{ix} / d_i)}{\sum_{i=1}^n f_i V_i / d_i} \quad (3-13)$$

$$Y_C^* = \frac{(\sum_{i=1}^n f_i V_i d_{iy} / d_i)}{\sum_{i=1}^n f_i V_i / d_i} \quad (3-14)$$

此时， X_C^* 和 Y_C^* 所对应的目标函数能取到最小值。其中，待建物流设施的坐标为 $C(X_C, Y_C)$ ，有 n 个配送中心、仓库或原材料供应地 D_i 的坐标分别为 (d_{ix}, d_{iy}) ， f_i 为待建物流设施到各处 i （配送中心、仓库或原材料供应地）的运输费率（单位产品运输单位距离的费用）， V_i 为运输到各处 i 的物流量， d_i 为工厂和各处 i 引入 d_x 后运输距离， TC 为总运输成本。

（2）重心法仓储中心选址模型的假设

需求量集中于某一点上，实际上需求来自于广阔区域内的多个销售门户。市场的重心通常被当作需求的聚集地，而这会导致某些计算误差，因为计算出的运输成本是到需求聚集地而不是到单个的需求点即建立的仓储中心。

（3）重心法仓储中心选址模型的建立

重心法首先要在坐标系中标出各个地点的位置，目的在于确定各点的相对距离、然后根据各点在坐标系中的横纵坐标值以各个销售门户的需求为输入，根据以下公式，通过 MATLAB 程序编写，求出每个划分好的区域市场需求的重心 (X_0, Y_0) ，并以此为建立仓储中心的坐标。若求解出的仓储中心坐标位置偏离备选位置，可首选临近销售需求量较大的门户，以此减少该区域的运输费用。

$$X_0 = \frac{\sum Q_i X_i}{\sum Q_i} \quad (3-15)$$

$$Y_0 = \frac{\sum Q_i Y_i}{\sum Q_i} \quad (3-16)$$

式中 X_0 --重心的 x 坐标, Y_0 --重心的 y 坐标, X_i --第 i 个销售门户的 x 坐标,
 Y_i --第 i 个销售门户的 y 坐标, Q_i --第 i 个销售门户的需求量。

(4) 重心法仓储中心选址模型的求解

根据以上公式, 编写 MATLAB 程序, 求解出各个区域的重心坐标如下表 3.8。

表 3.8 各区域重心坐标表

区域	序号	地区	x 坐标	y 坐标	需求 量	总需 求量	重心坐 标 x	重心坐 标 y
一	6	哈 尔 滨	2424	2292	5357	9603	2246.954	2110.448
	13	沈阳	2024	1881	4246			
二	3	成都	115	645	22523	45155	209.4565	489.3931
	5	贵阳	404	181	4260			
	8	昆明	0	0	5208			
	20	重庆	390	517	13164			
三	1	北京	1382	1641	19763	47754	1207.482	1428.299
	14	石 家 庄	1144	1471	2893			
	15	天津	1465	1555	10225			
	18	西安	616	1019	8792			
	19	郑州	1091	1097	6081			
四	2	长沙	1024	363	6364	26905	1101.044	301.9159
	4	广州	1069	197	8984			
	12	深圳	1156	-269	3454			
	17	武汉	1174	614	8103			
五	7	杭州	1709	624	16764	57523	1761.07	642.4906
	9	南京	1620	767	11567			
	10	宁波	1849	576	2892			
	11	上海	1867	663	21229			
	16	温州	1759	369	5071			

根据首选临近销售需求量较大的门户, 以此减少该区域的运输费用为基准, 选取各区域的仓储中心城市名称与位置坐标信息如下表 3.9。

表 3.9 各区域仓储中心坐标表

区域	序号	地区	仓储中心位置	x 坐标	y 坐标
一	6 13	哈尔滨 沈阳	长春	2288.58	2119.79
二	3 5 8 20	成都 贵阳 昆明 重庆	成都	115	645
三	1 14 15 18 19	北京 石家庄 天津 西安 郑州	石家庄	1144	1471
四	2 4 12 17	长沙 广州 深圳 武汉	长沙	1024	363
五	7 9 10 11 16	杭州 南京 宁波 上海 温州	常州	1734.62	756.22

由上述可知，该模型分为五个区域，并且每个区域根据重心法求得仓储中心的位置分别为长春、成都、石家庄、长沙和常州。由于每个区域的销售门户已知，因此每个区域的总销售量与各产品配比也可求得。所求得仓储中心的数量、位置、规模以及各产品配比如下表 3.10。

表 3.10 各仓储中心容量配比表

序号	仓储中心所在城市	总量	容	A 产品总需求量	B 产品总需求量	C 产品总需求量	A 占总容量比例	B 占总容量比例	C 占总容量比例
1	长春	9603		2355	3692	3604	24.40%	38.30%	37.30%
2	成都	45155		18922	11004	15302	41.80%	24.30%	33.90%
3	石家庄	47754		20029	10263	17611	41.80%	21.40%	36.80%
4	长沙	26905		10858	6977	9172	40.20%	25.80%	34.00%
5	常州	57523		19762	24946	12894	34.30%	43.30%	22.40%

由此可求出总成本 $G(x)$ 为 56.25 元。各仓储中心建造和维修成本以及总成本如下表 3.11。

表 3.11 总成本表

序号	仓储中心所在城市	总容量	每月建造与维修成本(万元)	总成本
1	长春	9603	4.3	56.25
2	成都	45155	13.3	
3	石家庄	47754	13.8	
4	长沙	26905	8.55	
5	常州	57523	16.3	

3.6 仓储中心选址模型小结

仓储中心在物流系统中具有极其重要的作用，也需要消耗大量的资源。因此，仓储中心的合理选址对于物流系统的规划起着至关重要的作用。在对仓储中心选址模型求解中，首先对 20 个销售门户进行了坐标化，将各个城市用二维坐标表示出以便进行系统聚类分析和重心法选址，其次用到 SPSS 软件对模型进行了聚类分析，将各个销售门户分成了五个大区。本模型将系统聚类和重心法相结合，提出基于最短距离法和重心法的配送中心选址流程，利用并设计重心法选址程序，有效地解决了多节点仓储中心的选址问题。重心法在单节点仓储中心选址问题时求解算法简单，选址的模型自由度比较高，但重心法作为单一配送中心选址方法是不妥当的，比如备选点是高山或湖泊。另外物流选址中需要考虑的因素很多：土地的可得性、交通是否便利、该地区城市发展方向等等。因此，重心法要与其他选址方法进行结合，才能在实际经营中发挥更大的作用。

4 问题三车辆调度模型

4.1 问题三重述

每个仓储中心建成后都需要一定数量的货车完成每月的运货任务，已知：

- a) 货车的采购成本为 15W 元，报废周期为 6 年，每月维护费用为 1000 元，运输费用为 150 元/百公里。
- b) 每辆货车可装载的最大容量为 250 单位产品。
- c) 每辆货车每天可行驶 500 公里。
- d) 保证每座城市在 7 天能获得一轮配货。

请在第 2 问确立的仓储中心的基础上，运用合理假设，建立模型，给出每个仓储中心需要配置的货车数和行驶路线，在成本最优的条件下满足常规的送货需求。

4.2 问题分析

物流配送优化主要是对配送车辆调度的一个优化问题，需要考虑多个约束条件并且同时达到多个目标，是一个复杂的组合优化，是一个难以解决的问题。车辆调度问题又可以分为单源调度问题 and 多源调度问题，我们这个是多源点向需求点配送货物的问题，相对于单源点配送要复杂的多。

本文在解决第二题仓储中心数量和选址的时候已经默认将所有的门店城市划分为 5 个区域，每个区域的仓储中心和门店城市已经确定，如此便将复杂的多源点配送问题转换为了相对比较简单的单源点配送问题。

本文将每个区域视为一个单源点配送的研究对象，把问题分成了四个阶段进行解决。第一阶段，采用动态聚类算法来给车辆分配派送任务，并得出每个区域需要的车辆运输批次，以及如果用一辆车给同一个区域的门店城市配送货物的话需要进出仓库几次。第二个阶段是用 Dijkstra 算法求出货车每条能达到最短里程的行驶路线，并求出相应的最短路线。第三个阶段是根据每个区域的配送路线总里程与每辆货车每周的最大行驶里程求出该区域需要安排的货车数量。第四阶段是根据每个区域的货车数量和每个区域货车需要行驶的总里程来

求总的运输成本，本文以一年为一个计费周期，并将一次性购买车辆的费用视为 6 年均摊。

问题假设

- (1) 各个仓储中心的货物供应量充足；
- (2) 每周向各门店运输一次货物且货车返回到仓储中心，即每年运输 52 次，以每年为一个费用统计周期；
- (3) 对于装载的货物，仅考虑数量的限制；
- (4) 门店城市每周的需求量是固定不变；
- (5) 货车一次性购买的费用六年均摊；
- (6) 每个门店城市的货物只由他所在区域的仓储中心配送。

公式符号一览表：

序号	符号	意义
1	A_i	第 i 个仓储中心
2	B_{ij}	第 i 个仓储中心的第 j 个门店城市
3	f_i	第 i 个仓储中心门店城市的个数
4	a_i	仓储中心 A_i 的存储量
5	x_{ij}	门店城市 B_{ij} 每周的需求
6	x_{ijt}	配送中心 A_i 派出车辆 t 为门店城市 B_{ij} 配送货物的量
7	y	每年的周数 52
8	c_1	运输费单价
9	c_2	每辆车每年的固定费用
10	c_3	购买一辆车的费用
11	c_4	一辆车每个月的维护费用
12	z	每辆车的额定载货量

13	l_{ij}	仓储中心 A_i 与门店城市 B_{ij} 的距离
14	l_i	仓储中心 A_i 所在区域的每周车辆的总里程
15	s	每辆货车每周的最大行驶量
16	a_{im}	区域 i 所需货车的数量
17	v_1	货车行驶的速度
18	k	所有区域每周的运输费用
19	M	所有仓储中心一年用于运输的总成本
20	x_{it}	配送中心 A_i 是否派出货车 t 执行配送任务，是取 1，否为 0
21	d_{ij}	门店城市 B_{ij} 需要整车次运输的车次
22	e_{ij}	门店城市 B_{ij} 整车次运输后的货物

4.3 建立模型

建立每个仓储中心需要配置的货车数及货车行驶路线的数学模型如下
所有仓储中心一年的总成本：

$$M = \sum_{i=1}^{i=5} \left(a_{im} \times ((c_3 \div 6) + c_4 \times 12) \right) + k \times y \quad (4-1)$$

各门店城市对仓储 A_i 的需求量小于或等于 A_i 的存储量：

$$\sum_{j=1}^{j=f_i} x_{ij} \leq a_i \quad (4-2)$$

每辆车装货不超过额定装载量：

$$\sum_{j=1}^{j=f_i} x_{itj} \leq 250 \quad (4-3)$$

每个配送中心指派的车辆数不超过其可调度的车辆总数：

$$\sum_{j=1}^{j=f_i} x_{it} \leq a_{im} \quad (4-4)$$

4.4 模型求解

4.4.1 启发式算法

根据模型特点，本文采用启发式算法，启发式算法描述如下：

步骤 1：由第二题可知每个门店城市所属的仓储中心。

步骤 2：考虑未分派车辆的所有配送中心，计算其地理位置重心。

步骤 3：对于门店城市内满足整车运输的货物量，优先从仓储中心派车，剩下不足整车的量按照步骤 4 计算。

步骤 4：从该区域中与重心距离最远的门店城市开始，以此门店城市为中心，添加相邻的门店城市，直至达到一辆车最大容量，把这些门店城市分派给同一辆车；同样的方法分配该区域的其它门店城市。若该群内的所有门店城市均分配完毕且满足所分派车辆的容量。得出每个区域各自需要车辆数以及每辆车分派的门店城市。

步骤 6：对每一辆车分派到的门店城市用 Dijkstra 算法求出该车的最短行驶路线。

步骤 7：求出总费用。

4.4.2 最短路径算法

本文通过汽车给区域内覆盖的城市运输货物，汽车在各个城市之间往返。由于本文中设计的模型，一个仓库给覆盖的区域城市超过两个以上，而且货仓所在的城市跟门店的城市不一定在一起，所这就要涉及到多个城市之间的相互往来，如果使得这个过程中产生最少的运费，那必须找到一条最短的往返的路径。常用的最短路径算法主要有，Dijkstra 算法，Floyd 算法，A*算法等，这些算法在求解路径最短问题有比较好的效果，但是跟本文中的应用场合并不完全吻合，因为本文的模型假设要求每个货仓覆盖的城市都必须要有供货，也就是每个点必须索引。

以区域 2 为例：

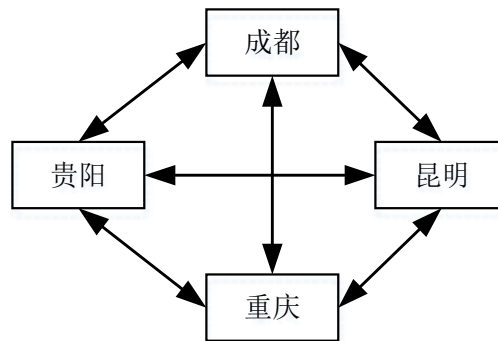


图 4.1 区域二配货线路

也就是从贵阳的仓库出发，要到达每一个城市，同样从任意一个城市出发也可以到达任何一个城市。这样每次遍历就要遍历到每个城市，但是访问的路线和先后顺序都是不同的，但是总会有一条路径最短的访问方式，这个问题就是著名的 TSP（旅行商）问题。

4.4.3 TSP 问题求解最短路径

TSP 问题的求解方法有很多种，比较经典求解方法有，贪婪算法、模拟退火算法，遗传算法，粒子群算法，蚁群算法等，这些算法的复杂度各不相同，在不同的问题上可能会得不到最优解。当 TSP 的节点比较多时，算法将十分的复杂，甚至无法求解，但是对于本文中的应用，节点比较少，而且对每个区域单独使用最优路径算法，这样最多的节点都不会超过十个，所以求解算法没有必要特别的复杂。

TSP 问题最简单的求解方法是枚举法。它的解是多维的、多局部极值的、趋于无穷大的复杂解的空间，搜索空间是 n 个点的所有排列的集合，大小为 $(n-1)!$ 。可以形象地把解空间看成是一个无穷大的丘陵地带，各山峰或山谷的高度即是问题的极值。求解 TSP，则是在此不能穷尽的丘陵地带中攀登以达到山顶或谷底的过程。

本文所使用的算法是贪心算法的思想求解 TSP 问题，虽然在节点较少的问题上可以采用枚举算法，但是本文为了使得求解过程比较快速，故而采用了贪心算法的思想求解 TSP 问题。

算法的步骤:

- 1、首先将每个城市作为一个节点，每个城市之间的距离作为权重，建立权重矩阵 G 。
- 2、将问题划分若干个子问题，分别对子问题进行求解。
- 3、从顶层开始逐次求解，每次求得一个子问题的最优解，也就是获取局部最短的路径。
- 4、记录每步计算得到的最优解以及每步遍历的节点编号。
- 5、检查所有子问题和所有节点，求解完子问题，然后将所有最优解合并成原问题的解，即获得城市的最短访问路径。

算法流程图:

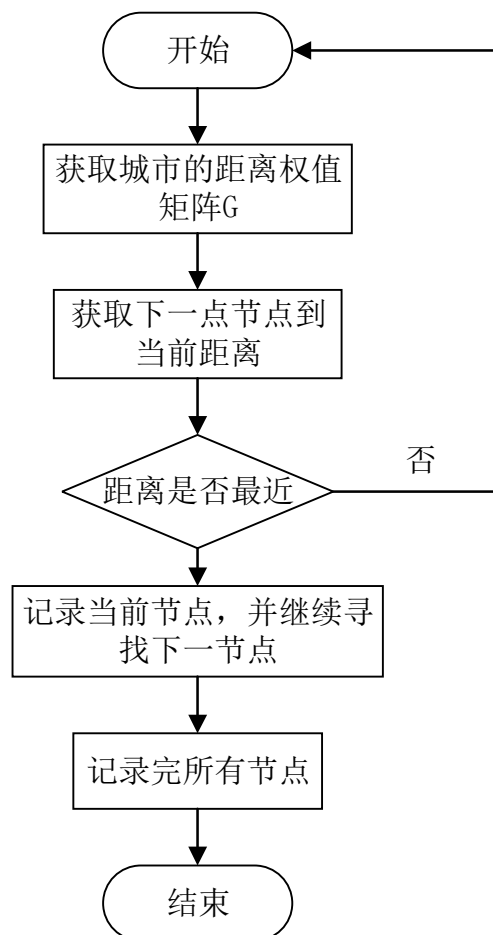


图 4.2 算法流程图

4.5 算例

4.5.1 算例描述

现为 20 个门店城市进行运输配送货物，已知这 20 个门店城市归属于五个不同的区域，并且这五个区域各自有一个仓储中心，已知货车到的采购成本为 15W 元，报废周期为 6 年，每月维护费用 1000 元，运输费用为 150 元/百公里；每辆货车可装载的最大容量为 250 单位，且每天最多可行驶 500 公里；另外需要保证每座门店城市每 7 天就获得一次配货。各个仓储中心及门店城市的坐标、仓储中心的车辆数、需求点的坐标所需货物的数量如表 1 所示试制定一个合理的运输方案使总的运输费用最少。

表 4-1 各门店城市坐标及月需求量

区域编号	序号	地区	x 坐标	y 坐标	月需求量
区域一	1	长春	2288.58	2119.79	0
	2	哈尔滨	2424	2292	447
	3	沈阳	2024	1881	354
区域二	1	成都	115	645	1877
	2	贵阳	404	181	355
	3	昆明	0	0	434
	4	重庆	390	517	1097
区域三	1	石家庄	1144	1471	242
	2	北京	1382	1641	1647
	3	天津	1465	1555	853
	4	西安	616	1019	733
	5	郑州	1091	1097	507
区域四	1	长沙	1024	363	531
	2	广州	1069	197	749
	3	深圳	1156	-269	288
	4	武汉	1174	614	676
区域五	1	常州	1734.62	756.22	0
	2	杭州	1709	624	1397
	3	南京	1620	767	964
	4	宁波	1849	576	241
	5	上海	1867	663	1770
	6	温州	1759	369	423

4.5.2 算例求解

(1) 为后续的计算需要，需要先求出每辆车每周的最大行驶里程。

每辆车速度 v_1 为 500 公里/每天，每周可运行 7 天，则他每周的最大行驶里程为 3500 公里。因每辆车每次的最大载货量为 250 单位，所以每辆车每周的最大运载能力为 875000。

(2) 为更好的为各区域分配货车以及指定最优的行车路线，需对各区域内的门店城市重新编号并以各城市间距离为因子建立矩阵如下：

区域一：包括长春、哈尔滨、沈阳三个城市，其中长春为单纯的仓储中心，哈尔滨和沈阳为门店城市，本文将三个城市之间的相互距离作为因子制成一个 3×3 的矩阵，如下所示，横纵坐标的顺序均依次为长春、哈尔滨、沈阳。

$$\begin{bmatrix} 0 & 360 & 315 \\ 360 & 0 & 675 \\ 315 & 675 & 0 \end{bmatrix} \quad (4-5)$$

依照区域一的方法，本文一次得到了区域二、三、四、五所有区域的距离矩阵以及各矩阵横纵坐标的顺序如下所示：

区域二： 1 成都 2 贵阳 3 昆明 4 重庆

$$\begin{bmatrix} 0 & 841 & 1094 & 340 \\ 841 & 0 & 714 & 511 \\ 1094 & 714 & 0 & 1092 \\ 340 & 511 & 1092 & 0 \end{bmatrix} \quad (4-6)$$

区域三： 1 石家庄 2 北京 3 天津 4 西安 5 郑州

$$\begin{bmatrix} 0 & 297 & 313 & 807 & 415 \\ 297 & 0 & 118 & 1224 & 722 \\ 313 & 118 & 0 & 1276 & 734 \\ 807 & 1224 & 1276 & 0 & 542 \\ 415 & 722 & 734 & 542 & 0 \end{bmatrix} \quad (4-7)$$

区域四： 1 长沙 2 广州 3 深圳 4 武汉

$$\begin{bmatrix} 0 & 662 & 994 & 392 \\ 662 & 0 & 142 & 991 \\ 994 & 142 & 0 & 1386 \\ 392 & 991 & 1386 & 0 \end{bmatrix} \quad (4-8)$$

区域五：1 常州 2 杭州 3 南京 4 宁波 5 上海 6 温州

$$\begin{bmatrix} 0 & 197 & 132 & 321 & 187 & 534 \\ 197 & 0 & 352 & 157 & 213 & 311 \\ 132 & 352 & 0 & 430 & 349 & 572 \\ 321 & 157 & 430 & 0 & 214 & 273 \\ 187 & 213 & 349 & 214 & 0 & 461 \\ 534 & 311 & 572 & 273 & 461 & 0 \end{bmatrix} \quad (4-9)$$

(3) 归属于仓储中心 A_i 城市门店 B_{ij} 每周的需求量 x_{ij} ，则该门店城市需要整车次运输的数量 d_{ij} ，余量 e_{ij} 如下：

$$d_{ij} = \lceil x_{ij} \div z \rceil \quad (4-10)$$

$$e_{ij} = x_{ij} - \lceil x_{ij} \div z \rceil \times z \quad (4-11)$$

表 4-2 各门店需要的整车次和余量

区域编号	序号	地区	月需求量	周需求量	整车次	余量
一	1	长春	0	0	0	0
	2	哈尔滨	447	103	0	103
	3	沈阳	354	82	0	82
二	1	成都	1877	434	0	0
	2	贵阳	355	82	0	82
	3	昆明	434	101	0	101
	4	重庆	1097	254	1	4
三	1	石家庄	242	56	0	56
	2	北京	1647	381	1	131
	3	天津	853	197	0	197
	4	西安	733	170	0	170
	5	郑州	507	117	0	117
四	1	长沙	531	123	0	123
	2	广州	749	173	0	173
	3	深圳	288	67	0	67
	4	武汉	676	156	0	156
五	1	常州	0	0	0	0
	2	杭州	1397	323	1	73
	3	南京	964	223	0	223
	4	宁波	241	56	0	56
	5	上海	1770	409	1	159
	6	温州	423	98	0	98

(4) 对每个区域中的各个城市整车运输后的余量进行车辆分配

从该区域中与重心距离最远的门店城市开始，以此门店城市为中心，添加相邻的门店城市，直至达到一辆车最大容量，把这些门店城市分派给同一辆车；同样的方法分配该区域的其它门店城市。若该群内的所有门店城市均分配完毕且满足所分派车辆的容量。得出每个区域各自需要车辆数以及每辆车分派的门店城市。得出每个区域的车辆配送如表 3-3 所示

(5) 对每一辆车分派到的门店城市用 Dijkstra 算法求出该车的最短行驶路线。

表 4-3 每个区域的车辆调度及最佳路线方案表

区域编号	门店城市	行车路线	里程/km	总里程/km
一	2、3	1-2-3-1	1350	1350
二	4	1-4-1	680	3339
	2、3、4	1-4-2-3-1	2659	
三	2	1-2-1	594	4398
	2、3	1-2-3-1	728	
	3、5	1-3-5-1	1462	
	4	1-4-1	1614	
四	2、3	1-2-3-1	1798	2582
	4	1-4-1	784	
五	3	1-3-1	264	1989
	4、6	1-4-6-1	1128	
	2、5	1-2-5-1	597	

(6) 配送中心 A_i 所需的总车辆为

$$a_{im} = \left[l_i \div s \right] + 1 \quad (4-12)$$

为了满足一般情况下常规的送货需求，本文每个区域求出的车辆数凡是小数的都用采用+1 取整的方式进行处理，最终得出仅区域三需要配置 2 辆货车，而其它区域均只需要 1 辆。

(7) 每年的总费用：

$$M = \sum_{i=1}^{i=5} a_{im} \times c_2 + \left(\sum_{i=1}^{i=5} l_i \div v_1 \times c_1 \right) \times y \quad (4-13)$$

由公式可以计算的，每年车辆购买和保养得固定费用为 222000 元，用于运输的费用为 213064.8 元，在最优的资源配置和路线规划下本文每年用在运输上的最优总成本为：435064.8 元。

4.6 结论

结论：本文为了满足常规下的送货需求，对每次计算出的货物需求量以及每个区域需要配置的货车数凡是带小数者均采用取整+1 的方式处理，这样虽然满足了送货需要，但也在无形中增加了成本预算。另外从计算结果可以发现，每个区域都会出现半车运输，就是运输货物的货车未装满的情况，造成了很大的资源浪费，所以，本文的模型还需要考虑进一步的优化，比如在规划区域的时候适当的减少区域个数，在一定范围内使更多的门店城市归属于同一个仓储中心，减少半车运输的情况，使成本更加优化。本文为了节约运输成本，在选择仓储中心的地址时尽可能选择距离重心最近的城市，如果重心靠近备选城市，则尽可能将仓储中心建立在门店城市所在的备选城市，可以最大程地降低运输成本。

5 总结与展望

本文针对仓储物流系统进行了产品销量预测、仓储中心地址以及车辆调度和最短路线规划分别进行了模型的分析、建立与求解。

本文所建立模型有以下优点：

（1）问题一中，预测模型采用了时间序列分解模型，能够分别对各种影响因素单独分析，很好的考虑到了商品销售时的淡旺季和长期趋势问题，这样能够得到更加真实的预测结果

（2）问题二中，仓储中心地址模型将系统聚类 and 重心法相结合，提出基于最短距离法和重心法的仓储中心选址流程，利用并设计重心法选址程序，有效地解决了多节点仓储中心的选址问题。将“在一个区域内建多个仓储中心的选址问题”转化为“在多个区域内分别进行单一仓储中心选址问题”，将求解模型简化，减少了大量计算。

（3）问题三中，车辆调度模型采用启发式算法，启发式算法具有原理简单、运行速度快、程序简单易修改等特点，通常被使用者接受，在实时管理中起到重要作用。

但是本文所建立模型依然有一些缺点：

（1）本文采用平面直角坐标系，但实际两个城市间的货物运输走的是公路里程，这样对求得的最佳仓储中心数量和地址产生影响，使本文得出来的方案与最优方案之间存在一定量的偏差。

（2）本文将问题二与问题三隔离开讨论，导致最终车辆配送时出现诸多半车配送的情况，造成了运输资源浪费，可以考虑根据每个城市的需求量与车载能力相结合来进行区域划分以及仓储中心的选择，如果能够将他们动态的链接起来形成一个反馈回路，互为输出输入效果会更加的优化。

综合看来，本文建立模型还需进一步改进：从模型角度，可以通过尝试其他算法模型，从而寻找最佳模型，提高程序的运行速度和模型的准确性，例如选择模拟退火、蚁群算法、遗传算法、免疫算法等智能算法。

参考文献

- [1] 袁磊磊. 基于时间序列分析的商业企业销售预测模型研究[D]. 重庆理工大学, 2013.
- [2] 关志民, 周宏波, 马钦海. 基于模糊多指标评价方法的配送中心选址优化决策[J]. 东北大学学报(自然科学版), 2005, 26(8):801-804.
- [3] 严冬梅. 城市物流中心选址问题研究[D]. 天津大学, 2004.
- [4] 韩中庚. 数学建模竞赛[M]. 科学出版社, 2007.
- [5] 唐元昭. 物流配送中心选址及车辆调度问题研究[D]. 浙江大学, 2011.
- [6] 吴翊. 数学建模的理论与实践[M]. 国防科技大学出版社, 1999.
- [7] 周根贵, 沈雁飞. 基于时间满意度的物流配送中心选址问题研究[J]. 浙江工业大学学报, 2008, 36(4):355-358.
- [8] 吴钦阳. 基于遗传算法的物流配送中心选址问题研究[J]. 福建电脑, 2009, 25(6):89-89.
- [9] 朱福庆. 启发式算法在物流配送中心选址模型中的应用[J]. 时代经贸, 2012(9):59-59.
- [10] 范李平. 物流配送及其车辆优化调度研究[D]. 上海海事大学, 2004.
- [11] 王志远. 车辆优化调度及物流配送管理系统模型的研究[D]. 大连交通大学, 2005.
- [12] 扶定, 卢兆成, 周国勤, 等. 系统聚类法在生态区域规划中的应用[J]. 安徽农业科学, 2003, 31(5):718-719.
- [13] 李婷婷, 黄晓东, 李珍萍. 多配送中心选址问题的数学模型及算法[J]. 物流技术, 2011, 30(15):85-88.
- [14] 蒋长兵, 王姗姗. 精确重心算法在物流节点选址中的应用[J]. 物流技术, 2005(9):65-67.

附录

附录一

	A商品预测结果											
城市	1	2	3	4	5	6	7	8	9	10	11	12
1	665	589	754	796	988	755	525	504	672	408	1012	544
2	225	213	228	160	172	169	101	120	81	59	47	24
3	797	1005	1078	801	867	888	637	981	821	741	733	886
4	351	364	404	414	391	418	340	342	380	336	401	411
5	162	214	235	139	195	131	117	135	170	103	158	96
6	111	138	155	132	133	102	82	137	120	110	122	122
7	579	444	465	348	702	413	238	438	214	243	478	207
8	250	206	216	158	118	191	152	148	161	118	106	169
9	686	401	385	529	671	467	300	497	328	463	482	452
10	88	129	116	104	153	123	64	97	100	67	116	91
11	313	296	565	659	811	1067	449	361	574	356	604	669
12	142	110	202	160	147	171	106	123	207	155	112	178
13	87	137	123	103	105	97	43	60	51	24	35	13
14	77	98	77	55	42	41	26	21	29	16	6	2
15	777	336	489	433	919	531	291	845	787	547	403	884
16	118	177	173	99	140	120	63	112	103	66	109	80
17	204	224	325	249	257	261	169	190	282	226	201	277
18	179	212	271	189	237	226	119	193	165	147	136	176
19	166	207	212	173	181	147	105	172	97	102	123	84
20	344	428	543	375	439	468	280	395	391	302	355	519

	B商品预测结果											
城市	1	2	3	4	5	6	7	8	9	10	11	12
1	252	274	226	333	224	192	90	114	111	109	54	24
2	158	172	188	125	158	145	94	134	135	125	178	154
3	410	651	586	425	426	529	320	602	476	463	417	647
4	112	140	135	109	120	130	73	106	112	102	116	159
5	49	82	84	57	76	49	36	51	72	53	88	48
6	152	208	199	196	149	169	122	246	207	255	219	234
7	217	375	1101	528	739	463	173	263	290	330	1208	1518
8	97	126	124	77	75	114	57	87	79	61	55	92
9	194	356	422	286	267	280	143	200	324	252	335	185
10	57	127	75	37	87	40	23	56	44	28	75	43
11	442	636	942	935	897	763	570	808	1334	1918	1074	1519
12	41	53	66	52	91	65	47	58	80	80	119	134
13	73	148	137	124	133	77	52	97	114	111	182	88
14	90	163	90	116	101	93	62	57	100	118	80	117
15	114	207	247	202	86	49	38	149	158	137	73	56
16	116	144	172	116	181	196	56	127	171	152	263	273
17	168	234	332	264	213	263	105	143	331	266	246	346
18	259	329	374	286	293	301	145	197	277	343	229	372
19	151	260	219	173	185	151	93	198	137	186	230	169
20	175	274	310	184	276	295	96	259	265	259	428	442

	C商品预测结果											
城市	1	2	3	4	5	6	7	8	9	10	11	12
1	417	528	841	738	1040	615	674	469	1064	668	1506	1003
2	156	210	231	198	249	269	123	235	270	249	385	442
3	300	450	501	333	390	563	249	554	645	572	713	1085
4	168	187	229	183	238	232	149	208	312	268	417	446
5	75	130	150	85	127	114	48	129	229	130	271	191
6	97	127	145	97	129	112	74	156	135	119	160	204
7	238	369	362	272	390	465	156	385	374	445	671	678
8	117	132	148	127	150	239	134	178	181	186	216	381
9	116	197	194	166	235	143	70	188	251	269	550	300
10	52	100	86	28	86	83	52	90	104	40	127	120
11	202	196	326	144	199	179	109	146	359	196	304	322
12	32	59	46	74	75	54	17	42	77	73	137	86
13	87	196	141	132	178	166	96	186	188	158	293	228
14	107	130	107	69	97	118	75	81	109	81	92	168
15	92	93	143	86	87	92	60	132	174	145	183	200
16	64	127	177	81	146	112	48	159	185	108	348	205
17	108	172	209	143	183	229	87	117	262	188	286	362
18	162	274	295	171	217	274	111	308	286	247	305	503
19	111	203	210	133	169	159	93	202	198	171	257	268
20	233	307	381	279	430	445	176	391	395	449	783	810

附录二

城市	A预测标准差	B预测标准差	C预测标准差
1	641.5365901	386.74726	146.851321
2	74.73644425	50.421164	35.0771189
3	207.314565	129.971086	73.6524631
4	97.43081981	40.0133114	32.7504647
5	52.46127132	33.968656	29.7006078
6	41.09441007	51.5770381	20.3585766
7	274.7027602	558.184933	50.7050005
8	44.26177657	36.1916124	23.9486403
9	329.153601	242.341246	53.9534521
10	46.77400758	44.3937524	20.3534932
11	498.0826689	295.378239	78.5957826
12	28.97514854	18.3961491	13.292393
13	66.04658962	64.8652967	32.693206
14	34.67313813	92.7475331	17.042268
15	291.5854493	118.796662	23.0933763
16	61.59251401	41.4596972	26.8716608
17	57.38721427	106.776004	36.1652843
18	64.54183114	110.903509	40.3522296
19	54.83043118	65.4936801	44.2214414
20	116.1986314	77.0098904	68.2766241

附录三

Sales_Forecast.m

```
clear;clc;
```

```
%% sales forecasting
```

```
sales_data = csvread('A 题数据附件 sales_data.csv',1,3);
```

```
[n,m] = size(sales_data);
```

```
city_num=20;Num=n/city_num;year =3;
```

```
A_gos=zeros(city_num,(Num/3));B_gos = A_gos;C_gos = A_gos; %每个城市的所有商品三年销售量
```

```
each_city = zeros(city_num,Num);
```

```
each_ct_yr = zeros(city_num,3*12);
```

```
%-----数据提取-----
```

```
for i=1:city_num
```

```
    each_city(i,:) = sales_data(((i-1)*Num+1):Num*i);
```

```
%        for j = 1:3
```

```
    A_gos(i,:) = each_city(i,1:36);
```

```
    B_gos(i,:) = each_city(i,37:72);
```

```
    C_gos(i,:) = each_city(i,73:108);
```

```

        each_ct_yr(i,1:12) = each_city(i,25:36);
        each_ct_yr(i,13:24) = each_city(i,61:72);
        each_ct_yr(i,25:36) = each_city(i,97:108);
    %    end
end
n_1 = 12;
for j = 1:36-n_1+1
    bm(j) = sum(A_gos(1,j:j+n_1-1))/n_1; %
    ym(j) = A_gos(1,6+j);
end
for i= 1:length(bm)-1
    TC(i) = (bm(i)+bm(i+1))/2;
    SI(i) = ym(i)/TC(i);
    % S(i) = SI(12-n_1+i)+SI()/year;
end
for i=1:n_1/2

    S(i)=(SI(n_1/2+i)+SI(n_1+i))/2;
    S(i+n_1/2)=(SI(i)+SI(n_1+i))/2;
end
S_1=[S(1:end),S(1:end),S(1:end)];
for i=1:36
    TCI(i) = A_gos(1,i)/S_1(i);
end
for j=1:20
    sumC(j) = sum(each_ct_yr(j,:));
end

% aa = sum(each_ct_yr);
% sum(aa)
% plot(1:36,TCI,'o','color',[rand(),rand(),rand()]);
% figure,plot(1:36,TCI,'color',[rand(),rand(),rand()],'LineWidth',1);
% figure,plot(1:36,A_gos(1,1:36),'color',[rand(),rand(),rand()],'LineWidth',1);
% s = sqrt(mean(ym(n)+1:m)-ym(1:end-1).^2);
% each_city(1,:) = sales_data(109:216);
% c = sales_data(109:110)

x=1:36;y=TCI;
px=1:48;
predic_TCI=zeros(12,1);
%-----regress 回归-----
% X = [ones(36,1) x' power(x',2) power(x',3) power(x',4) power(x',5)];
% X = [ones(36,1) x' power(x',2) power(x',3)];
X = [ones(36,1) x' power(x',2)];

```

```

[b,bint,r,rint,stats] = regress(y',X);
% rcoplot(r,rint)
z=b(1)+b(2)*px+b(3)*power(px,2);
plot(x',y','k+',px',z,'r','LineWidth',1.5);
xlabel('2014 - 2016 Month');ylabel('Quantity');
title('A 商品回归曲线（未消误差点）');
mask=(rint(:,1).*rint(:,2)<0);
ynew=y(mask);
Xnew=X(mask,:);
[b,bint,r,rint,stats] = regress(ynew',Xnew);
%
z=b(1)+b(2)*px+b(3)*power(px,2)+b(4)*power(px,3)+b(5)*power(px,4)+b(6)*power
(px,5);
% z=b(1)+b(2)*px+b(3)*power(px,2)+b(4)*power(px,3);
% z=b(1)+b(2)*px+b(3)*power(px,2);
% plot(x',y','k+',px',z,'r','LineWidth',1.5);
% xlabel('Month');ylabel('Quantity');
% title('C 商品回归预测曲线');
% plot(x',y','bo');
% xlabel('2014 - 2016 Month');ylabel('Quantity');
% title('A 商品消除季节因素后的销量散点图');
%-----多项式回归-----
% [y_1,s_1] = polyfit(x',y',5);
% % pred_y = polyval(y_1,px');
% Y=polyconf(y_1,px,S);
% plot(x,y,'k+',px,Y,'r');
% -----非线性回归-----

% beta0=[1 4]';
% [beta,r,J] = nlinfit(x',y','curvefun',beta0);
% [YY,delta] = nlpredci('curvefun',px',beta,r,J);
% plot(x,y,'b+',px,YY,'r');

% for i=1:12
% predicdata(i) = S(i)*predic_TCI(i);
% end

Quest_1.m
clear;clc;
%% all goods prediction
%-----获取数据-----
load A_gos.mat
load B_gos.mat
load C_gos.mat

```



```

n_1 = 12;
predic_TCI=zeros(12,3);
x=1:36;
px=1:48;
for cnt=1:20
    for j = 1:36-n_1+1
        A_bm(cnt,j) = sum(A_gos(cnt,j:j+n_1-1))/n_1; %
        A_ym(cnt,j) = A_gos(cnt,6+j);

        B_bm(cnt,j) = sum(B_gos(cnt,j:j+n_1-1))/n_1;
        B_ym(cnt,j) = B_gos(cnt,6+j);

        C_bm(cnt,j) = sum(C_gos(cnt,j:j+n_1-1))/n_1;
        C_ym(cnt,j) = C_gos(cnt,6+j);
    end
    for i= 1:length(A_bm)-1
        A_TC(cnt,i) = (A_bm(cnt,i)+A_bm(cnt,i+1))/2;
        A_SI(cnt,i) = A_ym(cnt,i)/A_TC(cnt,i);

        B_TC(cnt,i) = (B_bm(cnt,i)+B_bm(cnt,i+1))/2;
        B_SI(cnt,i) = B_ym(cnt,i)/B_TC(cnt,i);

        C_TC(cnt,i) = (C_bm(cnt,i)+C_bm(cnt,i+1))/2;
        C_SI(cnt,i) = C_ym(cnt,i)/C_TC(cnt,i);
    end
    for i=1:n_1/2
        A_S(cnt,i)=(A_SI(cnt,n_1/2+i)+A_SI(cnt,n_1+i))/2;
        A_S(cnt,i+n_1/2)=(A_SI(cnt,i)+A_SI(cnt,n_1+i))/2;

        B_S(cnt,i)=(B_SI(cnt,n_1/2+i)+B_SI(cnt,n_1+i))/2;
        B_S(cnt,i+n_1/2)=(B_SI(cnt,i)+B_SI(cnt,n_1+i))/2;

        C_S(cnt,i)=(C_SI(cnt,n_1/2+i)+C_SI(cnt,n_1+i))/2;
        C_S(cnt,i+n_1/2)=(C_SI(cnt,i)+C_SI(cnt,n_1+i))/2;
    end
    %
    AS_1(cnt,:)=[A_S(cnt,1:end),A_S(cnt,1:end),A_S(cnt,1:end)];
    BS_1(cnt,:)=[B_S(cnt,1:end),B_S(cnt,1:end),B_S(cnt,1:end)];
    CS_1(cnt,:)=[C_S(cnt,1:end),C_S(cnt,1:end),C_S(cnt,1:end)];
    for i=1:36
        %
        A_TCI(cnt,i) = A_gos(cnt,i)/AS_1(cnt,i);
        B_TCI(cnt,i) = B_gos(cnt,i)/BS_1(cnt,i);
        C_TCI(cnt,i) = C_gos(cnt,i)/CS_1(cnt,i);
    end
end

```

```

end
% sumC(j) = sum(each_ct_yr(j,:));
% [A_predit_TCI(cnt,:)] = calc_regress(x,px,A_TCI(cnt,:));
% B_predit_TCI(cnt,:) = calc_regress(x,px,B_TCI(cnt,:));
% C_predit_TCI(cnt,:) = calc_regress(x,px,C_TCI(cnt,:));
[A_predit_TCI(cnt,:),A_r(cnt,:)] = calc_regress(x,px,A_TCI(cnt,:));
[B_predit_TCI(cnt,:),B_r(cnt,:)] = calc_regress(x,px,B_TCI(cnt,:));
[C_predit_TCI(cnt,:),C_r(cnt,:)] = calc_regress(x,px,C_TCI(cnt,:));
for k = 1:12
    A_predit_data(cnt,k) = A_predit_TCI(cnt,36+k)*A_S(cnt,k);
    B_predit_data(cnt,k) = B_predit_TCI(cnt,36+k)*B_S(cnt,k);
    C_predit_data(cnt,k) = C_predit_TCI(cnt,36+k)*C_S(cnt,k);
end
city_predic(cnt,:) = sum(A_predit_data(cnt,:))+sum(B_predit_data(cnt,:))...
                    +sum(C_predit_data(cnt,:));
end
A_predit_data = ceil(abs(A_predit_data));
B_predit_data = ceil(abs(B_predit_data));
C_predit_data = ceil(abs(C_predit_data));
city_predic = ceil(city_predic);
%-----误差计算-----
% A_err=zeros(20,1);B_err=zeros(20,1);C_err=zeros(20,1);
for ct = 1:20
    A_err(ct) = sqrt(var(A_r(ct,:)));
    B_err(ct) = sqrt(var(B_r(ct,:)));
    C_err(ct) = sqrt(var(C_r(ct,:)));
End
Quest2.m
clc;
%% calc center of gravity

[Coordi,TXT,RAW]=xlsread('各城市坐标');
%-----Section_1-----
% asf = sum(city_predic);
ctmp_1 = 0;ctmp_2 = 0;
for ks = 1:2
    Coord_sec(1,ks) = (city_predic(6)*Coordi(6,ks)+city_predic(13)*Coordi(13,ks))/...
                    (city_predic(6)+city_predic(13));
%-----Section_2-----
ctmp_1=0;ctmp_2 = 0;
for ls = [3 5 8 20 ]

    ctmp_1 = ctmp_1+city_predic(ls)*Coordi(ls,ks);
    ctmp_2 = ctmp_2+city_predic(ls);

```

```

end
Coord_sec(2,ks) = ctmp_1/ctmp_2;

%-----Section_3-----
% Coord_sec(3,ks)
ctmp_1=0;ctmp_2 =0;
for ls = [1 14 15 18 19]

    ctmp_1 = ctmp_1+city_predic(ls)*Coordi(ls,ks);
    ctmp_2 = ctmp_2+city_predic(ls);
end
Coord_sec(3,ks) = ctmp_1/ctmp_2;
%-----Section_4-----
ctmp_1=0;ctmp_2 =0;
for ls = [2 4 12 17]

    ctmp_1 = ctmp_1+city_predic(ls)*Coordi(ls,ks);
    ctmp_2 = ctmp_2+city_predic(ls);
end
Coord_sec(4,ks) = ctmp_1/ctmp_2;
%-----Section_5-----
ctmp_1=0;ctmp_2 =0;
for ls = [7 9 10 11 16]

    ctmp_1 = ctmp_1+city_predic(ls)*Coordi(ls,ks);
    ctmp_2 = ctmp_2+city_predic(ls);
end
Coord_sec(5,ks) = ctmp_1/ctmp_2;
end

```

```

calc_regress.m
function [ z,r ] = calc_regress(x,px,y)
%
%
%-----regress-----
X = [ones(36,1) x' power(x',2)];
% X = [ones(36,1) x' power(x',2) power(x',3),power(x',4),power(x',5),power(x',6)];
[b,bint,r,rint,stats] = regress(y',X,0.05);
mask=(rint(:,1).*rint(:,2)<0);
ynew=y(mask);
Xnew=X(mask,:);
% rcoplot(r,rint)
z=b(1)+b(2)*px+b(3)*power(px,2);
%

```

```

z=b(1)+b(2)*px+b(3)*power(px,2)+b(4)*power(px,3)+b(5)*power(px,4)+b(6)*power
(px,5)+b(7)*power(px,6);
%-----
% [y_1,s_1] = polyfit(x',y',4);
% pred_y = polyval(y_1,px');
% Y=polyconf(y_1,x,S);
% plot(x,y,'k+',x,Y,'r');
%-----
% beta0=[1 4]';
% [beta,r,J] = nlinfit(x',y','curvefun',beta0);
% [z,delta] = nlpredci('curvefun',px',beta,r,J);
end

```

quest_3.m

```

clc;clear;

```

```

% W = [0 4 2 ;1 0 8;2 5 0];
% [ distance path] = Dijk( testw,0,1 )
% W=[0 50 inf 40 25 10
%     50 0 15 20 inf 25
%     inf 15 0 10 20 inf
%     40 20 10 0 10 25
%     25 inf 20 10 0 55
%     10 25 inf 25 55 0];
% [distance,path]=Dijk(W,2,5);
% distance,path
W_1 = [0 360 365;360 0 675;315 675 0];
W_2 = [0 841 1094 340;
       841 0 714 511 ;
       1094 714 0 1092;
       340 511 1092 0];
W_3 = [0 297 313 807 415
       297 0 118 1224 722
       313 115 0 1276 734
       807 1224 1276 0 542
       415 722 734 542 0 ];
W_4 = [0 622 994 392
       622 0 142 991
       994 142 0 1386
       392 991 1386 0
       ];
W_5 = [0 197 132 321 187 534
       197 0 352 157 213 311
       132 352 0 430 349 572

```

```

        321 157 430 0 214 273
        187 213 349 214 0 461
        534 311 572 273 461 0
    ];
    W_6=[0 321 534
          321 0 461
          534 461 0];
    W_7 = [0 197 187
            197 0 213
            187 213 0];
    inf=9999;
    % D=[inf 12 3 4 16;
    %     12 inf 7 8 9;
    %     3 7 inf 1 6;
    %     4 8 1 inf 3;
    %     16 9 6 3 inf];
    D = W_6;
    a=cell(1,length(D));
    i=1;
    a{1}=1;
    for k=2:length(D)
        D(:,i)=inf;
        b=min(D(i,:));
        i=find((D(i,:))==b);
        a{k}= i;
    end
end

```