数据分析·智慧模型

马钢国内外运输方案

2018年12月

目录

[绪论 3](#_Toc534224345)

[（一） 方案背景分析 3](#_Toc534224346)

[（二） 方案设计概述 4](#_Toc534224347)

[第一章 方案综述 纵观全局 7](#_Toc534224348)

[1.1数据分析——大数据分析建立数学模型 7](#_Toc534224349)

[1.1.1KNN模型 7](#_Toc534224350)

[1.1.2粒子群算法实现运量分配 8](#_Toc534224351)

[1.1.3均值漂移聚类分析 9](#_Toc534224352)

[1.2智慧模型——数据分析下的科学运输模型 10](#_Toc534224353)

[1.2.1长江运价指数模型 10](#_Toc534224354)

[1.2.2苏锡常杭地区铁路物流枢纽选址模型 10](#_Toc534224355)

[第二章 运价预测 运筹帷幄 13](#_Toc534224356)

[2.1长江运价分析 13](#_Toc534224357)

[2.1.1前言 13](#_Toc534224358)

[2.1.2方案背景分析 13](#_Toc534224359)

[2.1.3马钢现状及问题 16](#_Toc534224360)

[2.1.4长江运价指数模型 17](#_Toc534224361)

[2.1.6招标综合评价模型——层次分析法 24](#_Toc534224362)

[2.2外贸运价分析 26](#_Toc534224363)

[2.2.1前言 26](#_Toc534224364)

[2.2.2背景 26](#_Toc534224365)

[2.2.3数据分析—市场行情调查 27](#_Toc534224366)

[2.2.4数据处理—价格对照表 29](#_Toc534224367)

[2.2.5定性分析与航线分析 32](#_Toc534224368)

[2.2.6 SWOT分析选择贸易术语的策略 40](#_Toc534224369)

[2.3小结 40](#_Toc534224370)

[第三章 结构调整 提升比例 43](#_Toc534224371)

[3.1苏锡常杭地区铁路运输比例提升方案 43](#_Toc534224372)

[3.1.1前言 43](#_Toc534224373)

[3.1.2背景 43](#_Toc534224374)

[3.1.3运能安排现状及问题分析 50](#_Toc534224375)

[3.1.4铁路运输比例提升方案 52](#_Toc534224376)

[3.1.5可行性研究 62](#_Toc534224377)

[3.2小结 64](#_Toc534224378)

[总结 65](#_Toc534224379)

[参考文献 66](#_Toc534224380)

[附录 66](#_Toc534224381)

# 绪论

## 方案背景分析

根据国家统计局统计数据，2017年全国钢材产量10.6亿吨。产能过剩是我国钢铁产业目前面临的首要问题。根据中国钢铁工业协会相关统计数据，截至2018年4月，全国20个城市5大类品种钢材社会库存总量1252万吨。

运力严重过剩是航运业低迷的主要原因，针对这一问题，航运业的“供给侧改革”以去产能、去杠杆为核心思想和目标。通过自由竞争，优胜劣汰，在以市场为主导的前提下，充分发挥市场在资源配置中的核心作用，使得市场通过对供需的调节让货物对运力的需求量与市场中的船舶运力总量处于一个相对均衡的状态。还有，航运市场的基期运量改变，主要货物的流量流向明显改变，运价波动明显加大，影响因素增加。

伴随着供给侧改革推进，通过化解过剩产能、有效处置“僵尸企业”、淘汰落后产能、彻底取缔“地条钢”等措施，中国钢铁行业运营环境逐渐向公平竞争和绿色发展转变，中国钢铁企业经营情况显著好转。

据发改委、统计局及中物联调查，钢铁行业物流成本费用率为12.0%，仍然处于较高水平，比调查的全部工业企业高出2.2个百分点。若按调查的钢铁行业物流成本费用率水平推算，钢铁企业吨钢物流费用约为924元。其中，在物流成本构成中，运输成本占58.9%，仓储成本占2.9%，配送及物流加工成本分别占12.6%和7.8%。运输环节是控制物流成本的关键。目前来看，物流成本费用率与发达国家差距明显。以日本为例，日本钢铁企业物流费用率仅为4.9%，比中国低7.1个百分点。如果达到日本的水平，中国规模以上钢铁企业可节约3117亿元物流成本。中国钢铁物流成本远高于全球钢铁物流的平均成本，这一方面说明中国物流发展落后，物流成本高。同时也说明中国钢铁物流发展空间较大，利用现代物流提升中国钢铁产业和钢铁流通业的空间很大。

我国钢铁物流行业对于信息的整合能力有待加强，另外，钢铁行业面向的消费群体过于单一，使得钢铁物流行业的发展过于集中，只是采用现有资源的利用，使得钢铁物流企业的经营模式与时代相脱节，不利于行业的升级。我国是钢铁生产大国，同时也是物流运输大国，没有对现代化物流模式进行研究，在实际的工作中没有引入大数据，使得经济模式过于陈旧，不仅影响了区域内的经济发展，同时也抑制了钢铁产量的提升。

中国公路运输物流市场的规模高达4万亿，但传统的物流行业由于信息化技术落后导致信息不对称、货车空载返空率高、物流效率低等痛点，正困扰着这个万亿级市场的快速发展。物流成本居高不下，在“互联网+”时代背景之下，作为钢铁物流企业，更需要主动迎接“互联网+”所带来的供应链变革和商业创新的机遇与挑战。

通过调查我们发现，在国内外钢铁运输方面有一些突出问题：

航运指数中，上海航运交易所公布的数据针对的是沿海，而中联海运除了部分的沿海运输外，更多的是长江的内河运输，分为江海联运、江海直达等模式。针对长江领域的价格指数，并没有明确的参考价格，公司只能根据沿海运价指数对公司的航运指数进行上下的小幅度调整，但往往由于市场的延迟以及各种不确定因素的存在，长江运价与沿海运价有较大的差距。

中联海运涉及原材料和产成品的江海直达、长江内河运输等运输方式，目前并没有统一的长江运价指数可供参考，价格调整不能根据市场同步进行，具有相对的滞后性，对供应商体系的建立和完善也有很大的冲击。

近年来,随着国际贸易的开展以及国际航运条件和技术的改善，在我国对外达成的出口合同中, FOB术语的使用率越来越高排在各贸易术语的首位。以FOB术语成交的业务占80%以上，而且有逐步增长的趋势。FOB术语要进口商办理保险,那么出口商在自己的风险范围之内出现问题，难以第一时间去处理理赔事项，这会使我们在对外贸易中处于被动的地位，但还是有很多出口商为降低成本选择这一术语。那么在和外商签订合同之前，航线应该选择FOB的报价还是选择CIF报价成为钢铁出口的关键问题。

国家有关部门加强了对长江岸线港口码头专项整治、长江经济带饮用水水源地整治等系列治理行动，关停取缔了长江上近千座码头，对内河码头进行了集中整治。国家对内河码头综合整治将会造成一定时期内运力紧张，码头关停、航道限制等，这些都会压缩公司的船运物流能力，同时物流受阻、延期交货也会让客户不满意，影响到公司的信誉。所以必须要提前考虑调整运输结构，铁路运输虽然相比于水路运输成本较高，但和公路运输相比有着比较大的优势。

如果能对铁路运输路线进行优化布局，不仅可以减少运输时间，而且在运输安全上也会有很大的改善。另外，“前后一公里问题”会影响用户体验感，如何能在保证运输效率的情况下，提供“门到门”的全程物流运输服务，解决“前后一公里”的运输难题成为关键。

## 方案设计概述

基于发现的问题：

1. 长江领域的价格指数，没有明确的参考价格。
2. 没有统一的长江运价指数可供参考，价格调整不能根据市场同步进行，具有相对的滞后性。
3. 在和外商签订合同之前，航线应该选择FOB的报价还是选择CIF报价？
4. 国家对内河码头综合整治导致调整钢铁运输结构，如何提升铁路运输比例，优化铁路布局，解决“前后一公里”运输难题？

我们依据“数据分析·智慧策略”的理念，提出如下解决方案：

1. 针对没有统一的长江运价指数可供参考，我们提出长江运价分析方案，分析选取影响运价的主要因素，通过KNN算法建立长江运价指数模型，明确因素对长江运价指数的作用关系，模型具有能够实时地，准确地预测长江运价指数的优势，根据模型提出科学的调价运输策略，能有效降低招标成本，提高招标质量。
2. 针对航线应该选择FOB的报价还是选择CIF报价，我们提出外贸运价分析方案，定性分析FOB和CIF贸易术语对于航线的利弊，通过大数据分析建立数学模型，形成外贸价格对照体系，为马钢出口钢材提供科学的外贸运价策略。
3. 针对提高马钢在苏锡常杭地区的运输结构调整，我们提出苏锡常杭地区铁路运输比例提升方案，通过粒子群算法和聚类算法建立数学模型进行选址，在苏锡常杭地区布局一个或几个铁路物流枢纽，实现资源和物流要素的有效配置。将各装车点向临近运能较大的装车点集中，通过增设装卸线，改造发货点来提升各货运点的铁路发运能力，为苏锡常杭地区钢铁运输提供科学的运输策略。
4. 针对优化苏锡常杭地区铁路布局，解决“前后一公里”运输难题，我们提出多式联运方案。（待补充）

方案的总体设计思路如图0-1，本方案设计内容及运用的方法和工具见表0-1。



图0-1方案总体设计思路

表0-1本方案设计内容及运用的方法和工具



# 方案综述 纵观全局

## 1.1数据分析——大数据分析建立数学模型

### 1.1.1KNN模型

在回归（简单起见，这里讨论一元回归）领域，如果只知道某点的预测变量，要回归响应变量，只需要在横坐标轴上（因为不知道纵坐标的值，所以没法计算欧氏距离）选取K个最近的点，然后平均（也可以加权平均）这些点的响应值，作为该点的响应值即可。

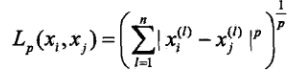
算法流程：

（1）根据给定的距离度量，在训练集中找出与样本x最近的k个点

（2）在这k个点中，根据分类决策原则（比如多数表决），决定x的类别y

特征有

（1）距离度量：特征空间中2个实例点的距离是2个实例点相似程度的反映



（2）K值的选择：K值较小使模型复杂化容易过拟合，K值较大使模型简单化容易欠拟合

（3）分类决策规则：一般是多数表决

根据KNN算法，列出构造KD树流程：

（1）现在有四维空间数据集https://img-blog.csdn.net/20150422123110047?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，其中https://img-blog.csdn.net/20150422123309144?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

（2）初始化根结点：以数据集T中所有实例点的坐标的中位数为切分点，将所有实例点分成3部分，坐标为中位数的落在根结点中，小于中位数的落在深度为1的左结点中，大于的落在深度为1的右结点中；

（3）重复：对于深度为 j 的结点，选择结点中所有实例点的坐标的中位数为切分点，将所有实例点分为3部分，坐标为中位数的落在 j 结点中，小于中位数的落在深度为 j+1 的左结点中，大于的落在深度为 j+1 的右结点中；其中https://img-blog.csdn.net/20150422123323442?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center。

（4）直到深度为 j 的结点的左右结点中没有实例点存在时，则结束KD树的划分；

最终整棵KD树的每一个结点都存放着一个K维数据样本点。

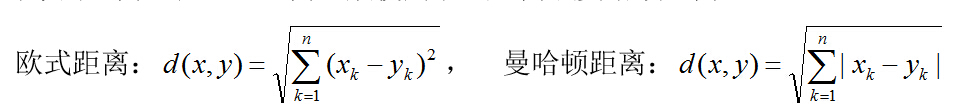
搜索KD树流程：

（1）在KD树中找出包含目标点 x 的叶结点：从根结点出发，递归地向下访问KD树，若目标点x当前维的坐标小于或等于切分点的坐标，则移动到左结点，否则移动到右结点，直到子结点为叶结点为止；

（2）以此叶结点为“当前最近点“；

（3）递归地往上回退，在每个结点进行以下操作：如果该结点保存的样本点比“当前最近点”距离目标点更近，则更新“当前最近点”；接着检查该结点的另一子结点对应的区域是否有更近的点，也就是检查另一子结点对应的区域是否与以目标点为球心，以目标点与”当前最近点“间的距离为半径的超球体相交（也就是结点的超平面与超球体是否相交）；如果相交，说明在另一个子结点对应的区域内有可能存在距目标点更近的点，则进入另一子结点区域进行搜索；

（4）当回退到根结点搜索结束后，“当前最近点”就是x的最邻近点；



### 1.1.2粒子群算法实现运量分配

（1）改进粒子群算法简介

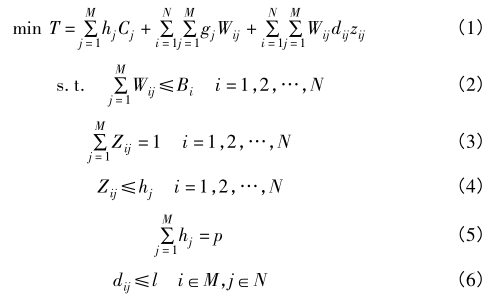
标准粒子群算法容易早熟收敛，容易陷入局部最优解。我们使用的改进算法通过引入领域均值，兼顾了粒子自身的认知思考、群体的全局最优位置和群体间的隐性知识，同时利用边界缓冲墙对越界的粒子进行缓冲，根据粒子的越界速度分情况进行处理，使算法的收敛速度和精度得到提高。

领域半径 R：在第t次迭代时粒子i的个体最优值为Pid(t)，所有粒子的全局最优值为 Pgd(t)，那么领域半径R为：



粒子群算法在寻优的过程中，粒子的位置和速度很有可能会突破其边界值，针对这种情况，采用边界缓冲墙的方法对越界粒子进行操作。

（2）选址模型建立



式(1)优化目标为总成本最小

式(2)表示分配至该站点的需求量小于等于站点规模容量

式(3)保证了每个码头将货物分配至离它最近的铁路站点

式(4)表示没有配送中心的地点不会有客户;

式(5)表示有 p个需求点被选为配送中心

式(6)表示配送中心只在限定的范围内对附近的需求点进行供应聚类算法

### 1.1.3均值漂移聚类分析

均值漂移聚类是基于滑动窗口的算法，来找到数据点的密集区域。这是一个基于质心的算法，通过将中心点的候选点更新为滑动窗口内点的均值来完成，来定位每个组/类的中心点。然后对这些候选窗口进行相似窗口进行去除，最终形成中心点集及相应的分组。不同于K-Means算法，基于密度的算法相比于K-Means受均值影响较小，并且均值漂移聚类算法不需要我们知道有多少类/组，有利于准确判断是否需新建铁路物流枢纽，同时准确确定枢纽位置、预估运量。

均值聚类求解步骤

① 确定滑动窗口半径r，以随机选取的中心点C半径为r的圆形滑动窗口开始滑动。均值漂移类似一种爬山算法，在每一次迭代中向密度更高的区域移动，直到收敛。

② 每一次滑动到新的区域，计算滑动窗口内的均值来作为中心点，滑动窗口内的点的数量为窗口内的密度。在每一次移动中，窗口会想密度更高的区域移动。

③ 移动窗口，计算窗口内的中心点以及窗口内的密度，知道没有方向在窗口内可以容纳更多的点，即一直移动到圆内密度不再增加为止。

④ 步骤一到三会产生很多个滑动窗口，当多个滑动窗口重叠时，保留包含最多点的窗口，然后根据数据点所在的滑动窗口进行聚类。

## 1.2智慧模型——数据分析下的科学运输模型

### 1.2.1长江运价指数模型

我们通过数据分析选取了影响长江运价的重要因素，分别是海运综合指数，煤炭指数，粮食指数，金属矿石指数，原油指数，成品油指数，然后用Python和C++语言利用KNN回归模型最终训练出了了长江运价指数模型（详细请见2.1长江运价分析），以下是我们拟合的结果（如图1-1）：

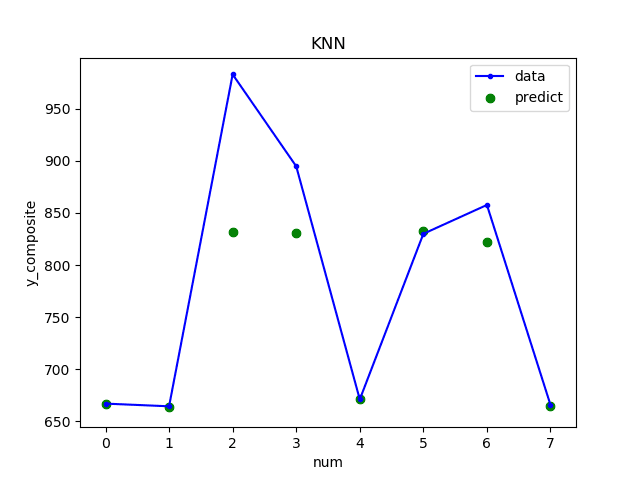
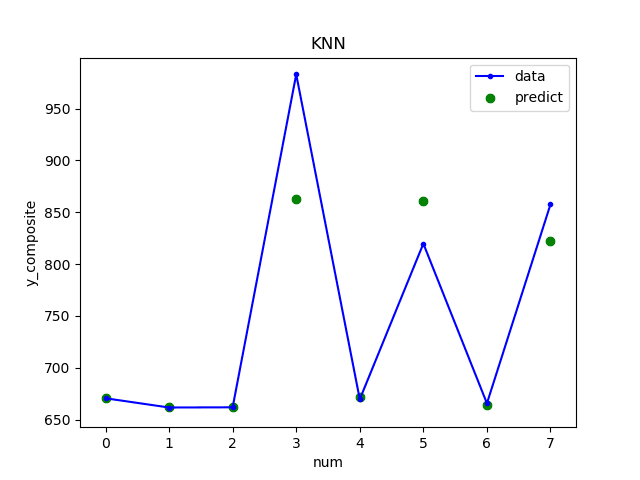


图1-1长江运价模型拟合结果图

至此，我们可以通过该实时预测模型，较为准确的预测长江运价指数。（详细代码见附件）

### 1.2.2苏锡常杭地区铁路物流枢纽选址模型

为满足马钢提升苏锡常杭地区铁路运输比例的需求，弥补铁路局现有铁路站点功能的不足，我们考虑针对水运情况与钢铁物流需求，通过粒子群模型算法及聚类算法进行选址，在苏锡常杭地区布局一个或几个铁路物流枢纽（详细请见3.1苏锡常杭地区铁路运输比例提升方案）。

根据水运总运量140万吨和现实码头运量的考虑，我们模拟了现实的数据以及模拟了铁路站点改造后的运量上限。

运用粒子群算法实现运量的初步分配

分配运量结果如图1-2所示：

图1-2粒子群算法分配运量图

可知需要新建铁路站点，杭州北和常州-石油分公司专用这两个站点承受运量过多，需要新建铁路站点来承受多出来的运量。

由上可知四个站点明显不够，需要新建站点。为此我们还利用了K-Means算法把码头分成了五类。由欧几里德距离作为最小化的依据。初步结果如图1-3所示：

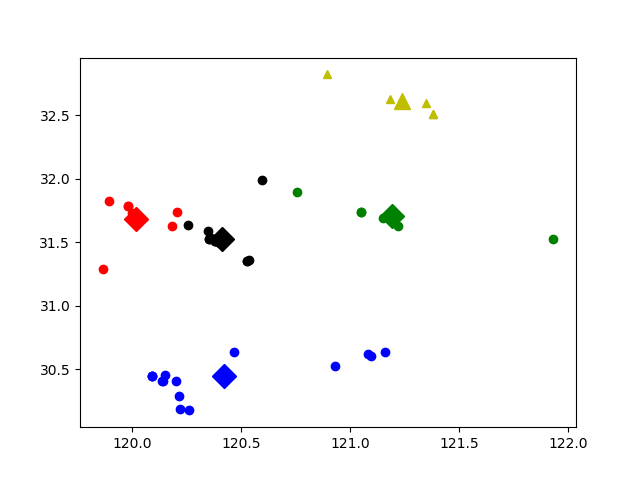


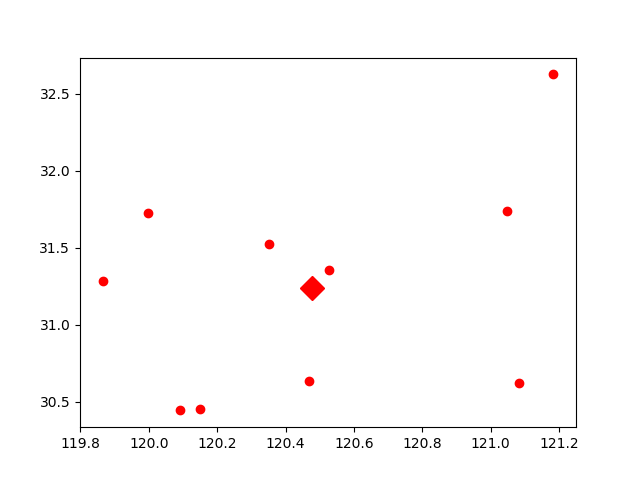
图1-3K-Means算法分类图

再根据已有的站点分析和码头运量分析哪些码头该分配到新站点

码头大多集中在苏州和无锡地区，所以我们决定新建站点在苏州附近

因为需要分配站点较少而且集中，所以本次模拟直接采用K-Means得出合理的选址范围，再加上适当的环境考量和分析。

K-Means结果如下：



再根据实际地理位置以及实际的情况考虑得出最终结果如下

①铁路沿线，不需要再新修铁路

②靠近主干道，靠近码头



# 运价预测 运筹帷幄

本章我们针对长江运价和外贸运价的相关案例和问题，提出了针对性的解决方案。在长江运价分析方面，我们通过数据分析建立了长江运价指数数学模型，在外贸分析方面，我们在数据分析的基础上提供科学的策略依据。科学的数学模型和数据分析可以帮助企业更好的参与竞标，进行组织规划等，进一步提高收益，降低运输成本，帮助决策者运筹帷幄之中，决胜千里之外。

## 2.1长江运价分析

### 2.1.1前言

航运指数中，上海航运交易所公布的数据针对的是沿海，而中联海运除了部分的沿海运输外，更多的是长江的内河运输，分为江海联运、江海直达等模式。针对长江领域的价格指数，并没有明确的参考价格，公司只能根据沿海运价指数对公司的航运指数进行上下的小幅度调整，但往往由于市场的延迟以及各种不确定因素的存在，长江运价与沿海运价有较大的差距。

针对没有统一的长江运价指数以及调整价格的滞后性等问题，我们通过背景分析、马钢现状分析以及数据分析，选取影响运价的主要因素，利用KNN算法建立了长江运价指数模型，能够实时地准确预测长江运价指数，同时根据模型的预测结果分析提出科学的调价策略供与参考。

### 2.1.2方案背景分析

#### 一、政策

新时期，党中央明确提出建立完善国家统计体系，长江航运指数占内河运价指数的70%，肩负着宏观上更好服务于市场供需关系监测、国民经济核算，微观上提供准确的运价信息，撮合市场运输交易的历史重任。

中华人民共和国交通运输部会以月为单位公布长江运价指数，并且会给出行情评述。长江航运运价指数自2007年运行以来，反映了长江干线主要货种的运价水平与变化，发挥着运输供需关系“晴雨表”作用。2012年根据《国内水运运输管理条例》加强市场监测的要求，指数编委会优化了运价调查的样本单位。2016年受环保督查和供给侧改革影响，长江航运市场发生新的变化，基于拉式方法确定的基期运量发生结构性变化，亟需作相应调整，重新设置权重。市场供给与需求发生变化，主要货物的流量流向明显改变，必需调整样本航线（调查的规格品）。运价波动明显加大，影响因素增加，综合指数与发布频次难以满足市场主体交易参考的需求。

#### 二、技术

2017年依据国家统计局《交通运输和邮政业价格统计报表制度》和交通运输部《关于推进交通运输统计改革全面提高统计数据真实性的实施意见》指示要求，在各级主管部门的指导下，运价指数编委会开展市场调研与研究，形成长江航运运价技术调整方案。

新方案主要调整内容：一是拉式链式公式替代基本拉式公式，适应长江航运市场复杂多变的发展形势。二是优化样本单位结构，吸纳更多的民营企业参与调查，增加了市场价格权重，改变原大型国企为主，协议价权重过大的局面。三是调整了煤炭、矿建材料等货类航线，优化了铁矿石、非金属矿等货类航线，集装箱航线。四是进一步优化航线样本单位，适时推出航线指数。五是应用移动互联网技术开发“长江航运指数调查与管理系统”，探索大数据、云平台技术应用，逐步提高发布频次，该系统于2018下半年上线运行。

上海航运交易所公布的数据涉及沿海的散货运输市场、国际干散货运输市场的价格走势，以及中国沿海散货、煤炭的运价指数。

#### 三、社会

1.港口非法竞争行为

部分港口的非法竞争行为，排除和限制了市场竞争，增加了实体经济的运行成本，也不利于港口自身做大做强，不能适应对外开放和经济发展的需要。

港口非法竞争的不利影响如表2-1。

表2-1港口非法竞争的不利影响

|  |  |
| --- | --- |
| 不利影响 | 内容 |
| 增加了实体经济的运行成本 | 一是港口的限定交易行为，使得相关市场完全缺乏有效竞争，服务价格高。二是对本地集装箱收取的装卸费用过高。这部分费用由港口直接向船公司收取，再由船公司向货主收取，费用过高也增加了实体经济的负担。三是强制要求交易对象接受一些不必要的服务。 |
| 影响了公平透明的营商环境 | 港口是国家重要的基础设施，也是一个复杂的业务系统。有的业务只能由港口完成，有的业务已经引入市场竞争。很多港口在已经形成竞争的业务上，只允许自己下属企业提供服务，而不允许其他社会资本进入，人为制造了市场壁垒，没有形成对各类市场主体“权利平等、机会平等、规则平等”的准入环境。 |
| 不利于港口自身做大做强 | 竞争是企业发展的根本动力。只有在竞争的压力下，企业才有动力积极创新，提升生产经营效率。港口与交易对象签署忠诚条款，甚至是不竞争条款，以此来逃避市场竞争。这种做法虽然在短期内可以维护自身的市场地位，获取更高收益，但从长远看，势必会削弱企业创新发展的动力，不利于企业自身做大做强，也会降低经济运行的效率。 |

2.长江航运存在的问题

伴随着长江经济带建设工作的推进，在各项有利因素凸显的同时，各种不利因素也接踵而来，这些问题若得不到很好的解决势必会影响长江航运物流产业的进一步发展。长江航运存在的问题见表2-2。

表2-2长江航运存在的问题

|  |  |
| --- | --- |
| 问题 | 内容 |
| 物流成本过高 | 物流运输中各种收费太多，包括道路的收费、管理的收费、办理各种证件的收费等等。其次，物流设施落后，基础设施的落后直接导致运输成本上升，降低了流通效率。再次，物流的管理体制、运行机制不成熟。 |
| 物流体系不健全 | 我国并没有完善统一的物流体系，铁路、公路、水运构成了主要的运输方式，以国企央企为核心的物流巨头承担了大部分的运送任务，物流行业从整体规划到区域布局都存在着重大问题 |
| 各种运输方式没能有效配合 | 干线和直线运输道路、龙头企业和中小民企之间无法有效对接，使得运送效率极低、运输成本攀升；而且在另一方面，关卡太多、收费太贵是导致物流成本高企的直接因素，管理较差、沟通成本高也在一定程度上增加了整体运输费用。 |

### 2.1.3马钢现状及问题

进口矿市场的运价与市场相对有很大差异，对价格的调整滞后比较严重，煤焦的价格走势相对乐观，每年马钢的运量又比较大，单位价格的差异，带来的损失是巨大的。究其原因是没有统一的长江运价指数参考体系，对市场的预测有一定程度的延缓，以及不确定性因素的影响：洪水、台风等恶劣天气导致船舶压港情况严重，周转不畅使得市场有效运力减少，进而推动运输价格迅猛上涨；油价的波动，会影响船舶的使用费用，在市场的驱动下，成本波动必然导致运价波动；沙石受国家环保政策的影响，在供应和需求不平衡的情况下，进行沙石贸易会获利更多，造成船舶向沙石运输倾斜，从而影响煤炭、铁矿石等的运输，导致运价波动；沿海运输的繁忙程度影响沿江的运输量，受需求拉动的影响，运价也会在一定范围内波动。因此如何快速响应市场，确定合理的调价幅度是需要解决的重点。

近几年，航运业逐渐复苏，大宗商品的运输价格走势也需要重新界定。作为由马钢物流公司控股62%的航运企业，安徽中联海运有限公司主要从事国内沿海、长江、内河运输。通过合作一批江海运输、长江运输、内河运输的船舶，形成江、海运力规模超过30万吨以上。

每一次航运定价尤为重要，建立一个权威性、有说服力的价格体系，是需要解决的关键。而航运运价指数反映了航运市场的价格水平和动态，使航贸相关人士及时地了解市场供求状况，为企业经营决策提供依据，同时也有助于政府主管部门的宏观决策与调控。

从案例中可知，中联海运涉及原材料和产成品的江海直达、长江内河运输等运输方式，目前并没有统一的长江运价指数可供参考，价格调整不能根据市场同步进行，具有相对的滞后性，对供应商体系的建立和完善也有很大的冲击。长江运价问题主要体现在运价指数和调价两方面。（见表2-3）

表2-3长江运价现存问题

|  |  |
| --- | --- |
| 主要问题 | 问题因素 |
| 运价指数 | 上海航运交易所公布的数据针对的是沿海，目前没有统一的长江运价指数可供参考。 |
| 调价 | 公司只能根据沿海运价指数对公司的航运指数进行上下的小幅度调整，但往往由于市场的延迟以及各种不确定因素的存在，长江运价与沿海运价有较大的差距。 |

### 2.1.4长江运价指数模型

#### 2.1.4.1数据来源

中国交通运输部数据如下表：



表2-4中国交通部数据

#### 2.1.4.2单因素分析模型

根据交通运输部网站上提供的相关数据信息，我们提取了六个因素作为备选因素进行分析。他们分别是海运综合指数，煤炭指数，粮食指数，金属矿石指数，原油指数，成品油指数。近三年的数据数值化以后可得

我们提取了六个备选因素后对其进行了归一化处理，以消除值域对其的影响。

根据通常的手段，我们可以建立单因素分析模型。

该函数是建立x到y的映射关系。根据残差理论和通用误差分析式可得

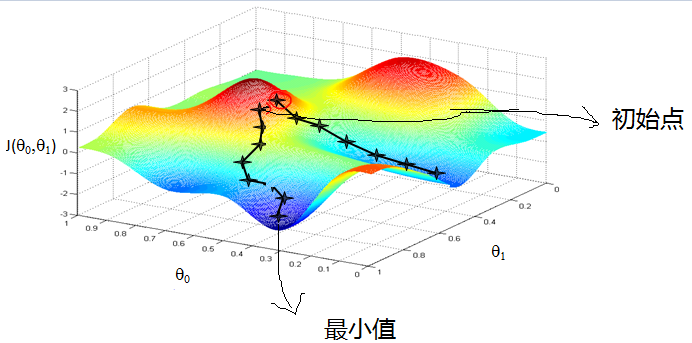
简单分析该误差值极值取到的条件

##### (i)线性拟合

参数化映射关系

实例化线性拟合误差关系

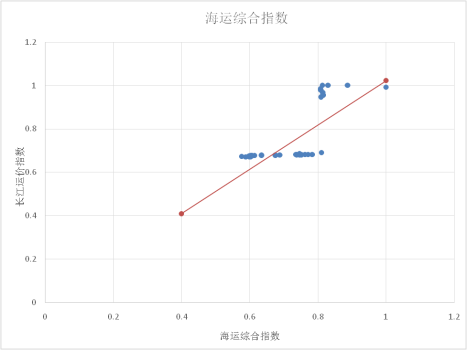
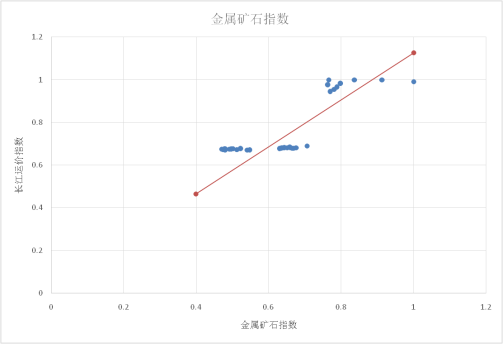
梯度下降算法：在机器学习算法中，在最小化损失函数时，可以通过梯度下降法来一步步的迭代求解，得到最小化的损失函数，和模型参数值。比如我们在一座大山上的某处位置，由于我们不知道怎么下山，于是决定走一步算一步，也就是在每走到一个位置的时候，求解当前位置的梯度，沿着梯度的负方向，也就是当前最陡峭的位置向下走一步，然后继续求解当前位置梯度，向这一步所在位置沿着最陡峭最易下山的位置走一步。这样一步步的走下去，一直走到觉得我们已经到了山脚。当然我们也有可能走到山谷，这就需要多次随机初始值来保证求出的是最优解。



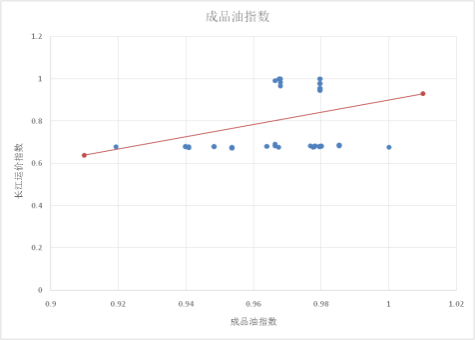
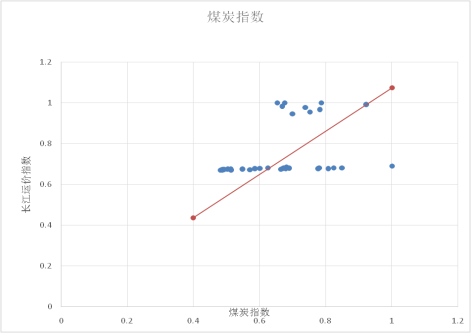
将该算法运用于该模型，获得参数化公式：

我们用c++11实现了该算法（见附件），并得到了六个因素的拟合结果，可视化操作下结果如下：

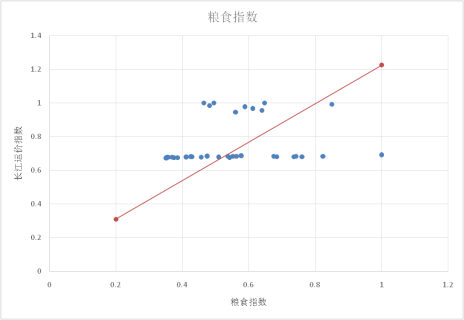
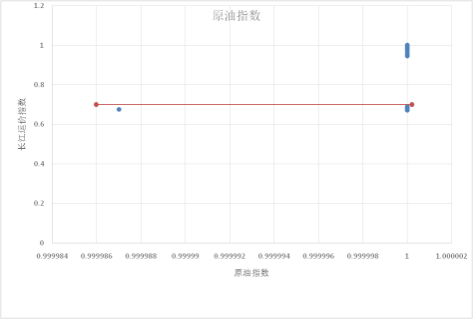
|  |  |
| --- | --- |
| 因素名称 | 平均误差系数 |
| 海运综合指数 | 0.00442 |
| 煤炭指数 | 0.01862 |
| 粮食指数 | 0.03634 |
| 金属矿石指数 | 0.0072 |
| 原油指数 | 0.0992 |
| 成品油指数 | 0.014212 |

I（海运综合指数）=0.00442 I（金属矿石指数）=0.0072

I（成品油指数）=0.014212 I（煤炭指数）=0.01862

I（粮食指数）=0.03634 I（原油指数）=0.0992

图2-1因素拟合结果

##### (ii)非线性拟合——最小二乘法非线性演化

建立模型

通过线性拟合的推导公式，我们可以获得如下计算公式

进行非线性演化

用矩阵形式表达可以得到

矩阵表示的最优化函数为

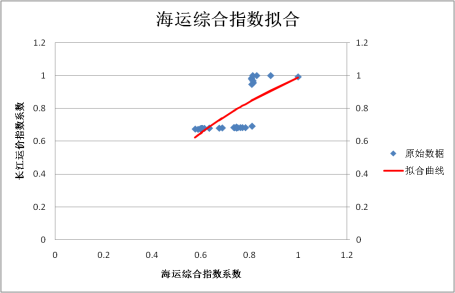
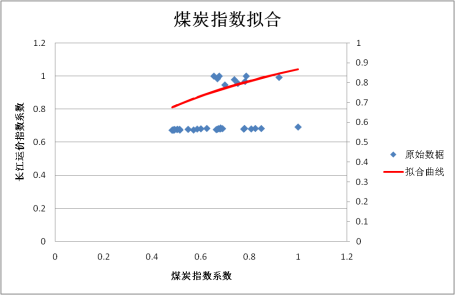
其中

对进行求导并令其为0，可得

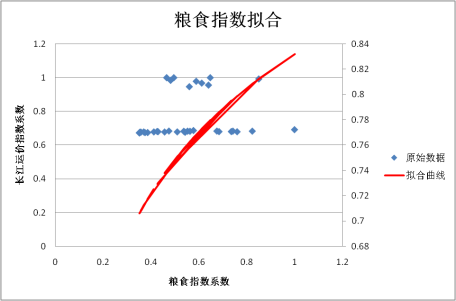
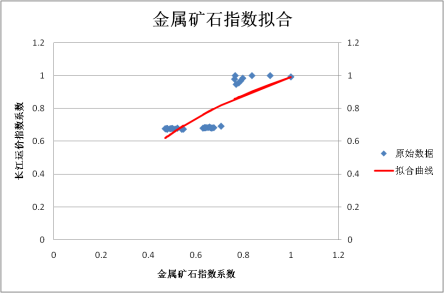
最终可得最优解

通过excel和c++我们最终得到误差值

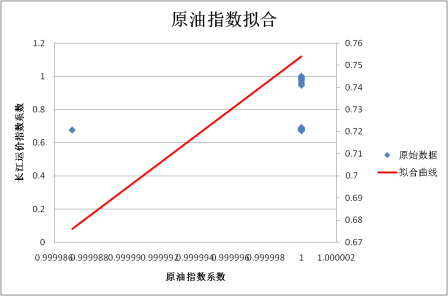
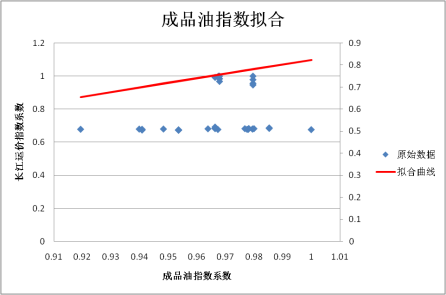
|  |  |
| --- | --- |
| 因素名称 | 平均误差系数 |
| 海运综合指数 | 0.0077 |
| 煤炭指数 | 0.0141 |
| 粮食指数 | 0.0157 |
| 金属矿石指数 | 0.0061 |
| 原油指数 | 0.0167 |
| 成品油指数 | 0.0157 |

I（海运综合指数）=0.0077 I（煤炭指数）=0.0141

I（粮食指数）=0.0157 I（金属矿石指数）=0.0061

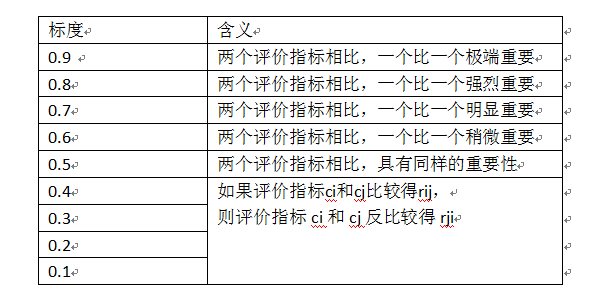
I（原油指数）=0.0167 I（成品油指数）=0.0157

图2-2指数拟合结果

#### 2.1.4.3.基于单因素分析模型的多因素分析模型

##### (i)多因素的选取

多因素比较方法fuzzy矩阵



表示相对重要度,从单因素模型中获得

下面为多因素模型的fuzzy矩阵

根据矩阵我们选出了海运综合指数、煤炭指数、金属矿石指数、成品油指数作为多因素模型的参数变量。

##### (ii)KNN模型的建立

在回归（简单起见，这里讨论一元回归）领域，如果只知道某点的预测变量，要回归响应变量，只需要在横坐标轴上（因为不知道纵坐标的值，所以没法计算欧氏距离）选取K个最近的点，然后平均（也可以加权平均）这些点的响应值，作为该点的响应值即可。

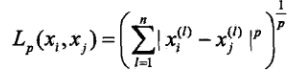
算法流程：

（1）根据给定的距离度量，在训练集中找出与样本x最近的k个点

（2）在这k个点中，根据分类决策原则（比如多数表决），决定x的类别y

特征有

（1）距离度量：特征空间中2个实例点的距离是2个实例点相似程度的反映



（2）K值的选择：K值较小使模型复杂化容易过拟合，K值较大使模型简单化容易欠拟合

（3）分类决策规则：一般是多数表决

根据KNN算法，列出构造KD树流程：

（1）现在有四维空间数据集https://img-blog.csdn.net/20150422123110047?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，其中https://img-blog.csdn.net/20150422123309144?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

（2）初始化根结点：以数据集T中所有实例点的坐标的中位数为切分点，将所有实例点分成3部分，坐标为中位数的落在根结点中，小于中位数的落在深度为1的左结点中，大于的落在深度为1的右结点中；

（3）重复：对于深度为 j 的结点，选择结点中所有实例点的坐标的中位数为切分点，将所有实例点分为3部分，坐标为中位数的落在 j 结点中，小于中位数的落在深度为 j+1 的左结点中，大于的落在深度为 j+1 的右结点中；其中https://img-blog.csdn.net/20150422123323442?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZHJlYW1fY2F0Y2hlcl8xMA==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center。

（4）直到深度为 j 的结点的左右结点中没有实例点存在时，则结束KD树的划分；

最终整棵KD树的每一个结点都存放着一个K维数据样本点。

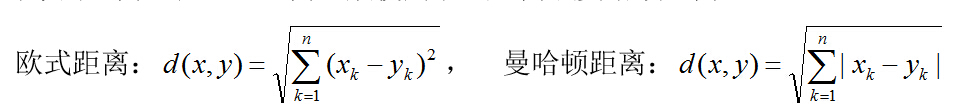
搜索KD树流程：

（1）在KD树中找出包含目标点 x 的叶结点：从根结点出发，递归地向下访问KD树，若目标点x当前维的坐标小于或等于切分点的坐标，则移动到左结点，否则移动到右结点，直到子结点为叶结点为止；

（2）以此叶结点为“当前最近点“；

（3）递归地往上回退，在每个结点进行以下操作：如果该结点保存的样本点比“当前最近点”距离目标点更近，则更新“当前最近点”；接着检查该结点的另一子结点对应的区域是否有更近的点，也就是检查另一子结点对应的区域是否与以目标点为球心，以目标点与”当前最近点“间的距离为半径的超球体相交（也就是结点的超平面与超球体是否相交）；如果相交，说明在另一个子结点对应的区域内有可能存在距目标点更近的点，则进入另一子结点区域进行搜索；

（4）当回退到根结点搜索结束后，“当前最近点”就是x的最邻近点；



我们采用了欧式距离作为距离的度量方法，最终训练出了模型，下图展示了我们训练的拟合结果。

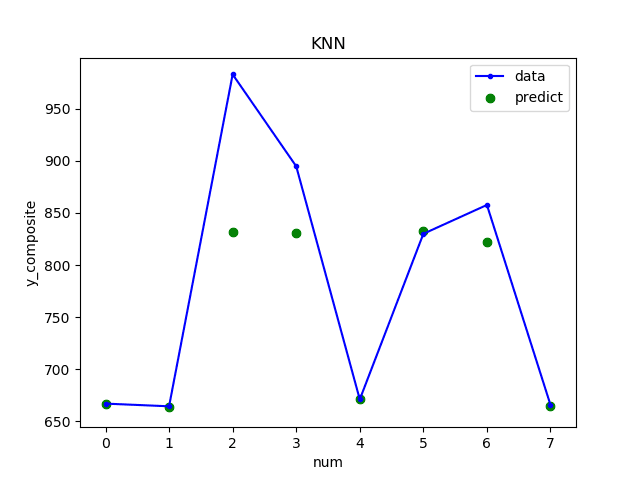
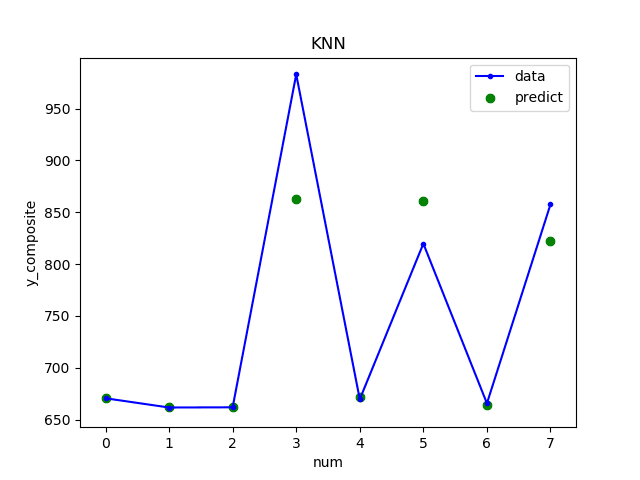


图2-3长江运价指数模型拟合结果

至此，我们可以通过该实时预测模型，较为准确的预测长江运价指数。（详细代码见附件）

### 2.1.6招标综合评价模型——层次分析法

评价模型

流程

价格

安全性

效率

稳定性

其他因素

供应商

各因素评价指数

其中表示每一个专家的评价指数，由专家评估所得。

模型综合评价指数

为各因素评价指数，为各指数的权重系数

从中，我们可以评价长江运价指数的实时预测对最终招标产生的优化效果。

我们可以用该评价模型来评价我们上述的预测方法

## 2.2外贸运价分析

### 2.2.1前言

现在国际贸易部和外商签订合同前就已经确定CIF和FOB，但是在接受国际贸易部的指令后，在市场上找船，是以比价的形式进行招标，以最低价中标。和外商签订合同之前，航线应该选择FOB报价还是选择CIF报价，哪一个更适合马钢的航线，都是马钢亟待解决的问题。

我们通过数据分析与数据处理，对当前市场行情做了充分调研，并建立了价格对照体系。定性比较选择FOB与CIF报价对于出口商的利弊；分析对于某一条或几条航线，客户选择FOB的原因，以及选择CIF对客户和马钢物流的利弊情况；归纳航线选择FOB报价或者CIF报价的优缺点。

### 2.2.2背景

**（1）FOB术语在我国出口业务中的使用**

随着我国航运市场改革的深入和逐步开放，外国班轮公司迅速进入我国航运市场开展货运代理业务。由于外国班轮公司、货代企业服务质量好、价格优惠,进口方都愿意指定外国班轮公司和货代安排运输，因此会要求签署FOB合同条款；同时，由于国际运输市场运力紧张，油价不断上涨导致运价频频上调,加上各种各样的附加费,对运输行业价格的波动,各出口公司难以准确控制。为了便于外贸成本核算,不得不选择FOB条款,考虑到有效规避对外贸易中的不确定性，因此，导致我国出口企业使用FOB术语成交的外贸合同比例迅速增长。

**（2）CIF术语在我国出口业务中的使用**

由于当今国际贸易中存在许多的不确定因素，进口商在贸易术语的选择中更趋向选择风险小、责任少的术语，进而大大发展了CIF术语在国际贸易进口业务中的使用比例。目前在进口业务中，我国进口商为了规避贸易中一些不必要的风险，也越来越多的采用了CIF贸易术语进行国际业务往来，导致CIF贸易术语在我国进口业务中的使用频率逐年提高，上升为我国对外进口合同中使用频率最频繁的贸易术语。

**（3）国际贸易的选择**

FOB是当前国际贸易最常用的条款，原因有二。其一是外商倾向于选择自己熟悉且对自己有利的一方，其二是卖家想避免拖车、驳船、报关、退税等一系列业务。

### 2.2.3数据分析—市场行情调查

#### 2.2.3.1FOB与CIF概述

表2-4 FOB和CIF的外贸报价方式的汇总

|  |  |  |
| --- | --- | --- |
| 报价方式 | CIF | FOB |
| 交货地点 | 装运港 | 装运港 |
| 运输 | 卖方负责 | 外商负责 |
| 保险 | 卖方负责 | 外商负责 |
| 出口手续 | 卖方负责 | 卖方负责 |
| 进口手续 | 卖方负责 | 外商负责 |
| 风险转移 | 装运港船舷 | 装运港船舷 |
| 所有权转移 | 随交单转移 | 随交单转移 |

1. 简介

FOB贸易条款即船上交货。它指卖方在指定的装运港把货物送过船舷后交付，货过船舷后买方须承担货物的全部费用、风险、灭失或损坏，另外要求卖方办理货物的出口结关手续。

CIF贸易条款即卖方必须支付将货物运至指定的目的港所需的运费和费用，但交货后货物灭失或损坏的风险及由于各种事件造成的任何额外费用即由卖方转移到买方。但是，在CIF条件下，卖方还必须办理买方货物在运输途中灭失或损坏风险的海运保险。因此，由卖方订立保险合同并支付保险费。买方应注意到，CIF术语只要求卖方投保最低限度的保险险别。如买方需要更高的保险险别，则需要与卖方明确地达成协议，或者自行作出额外的保险安排。

1. 区别

表2-5 FOB和CIF的区别

|  |  |  |
| --- | --- | --- |
| 区别 | FOB | CIF |
| 港口性质 | FOB后的港口指卖方所在国的港口 | CIF后的港口指买方所在国的港口 |
| 费用构成 | FOB是指成本价，也叫离岸价。  =货物的出厂价+出口报关货物装到买方指定船舱中产生的一切费用+自己的利润 | =货物的出厂价+出口报关货物装到买方指定船舱中产生的一切费用+自己的利润+海运费+保险费 |
| 保险费的支付和办理 | FOB保险由买方办理，卖方应于装船前通知买方 | CIF保险由卖方办理并支付保险费，卖方按合同条款，保险条款办理保险并将保险单交给买方 |
| 租船订船 | FOB价格由买方指定船公司/船代公司甚至货代公司安排船运，买方能否及时租船订舱，会影响卖方的及时交货以及银行交单等 | CIF由卖方自主选择船公司或货代公司 |
| 装船通知告知买方时间 | FOB规定装船通知应在装船前告知买方，包括装船内容、装船细节以便买方有充足的时间办理货物海上保险 | CIF是由卖方投保，可在装船后几天内告知买方装船通知 |
| 承担不可抗力风险 | 买方办理保险，保单在买方手里，保险公司又大多在国外，卖方难以向保险公司索赔 | 卖方办理保险，在启运港投保，在客户拒付退单的情况下，卖方可凭保单向当地保险公司索赔 |

**2.2.3.2FOB与CIF优劣对比**

表2-6 FOB与CIF优劣对比

|  |  |  |
| --- | --- | --- |
| 贸易术语 | 优点 | 缺点 |
| FOB | 1. 在运费和保险费波动不稳的市场条件下于出口商有利 2. 可增进社会福利 3. 流程简单 | 1. 订舱出运紧张 2. 有较大风险 3. 收益低 4. 操作麻烦 5. 对国轮、保险业发展不利 |
| CIF | 1. 实时关注生产进度，在货物生产之前尽早订船，减少资金的占用。 2. 掌握对外出运的主动权 3. 价量大的外贸企业可以集中运量与船公司商谈优惠运价 4. 减少了FOB价指定货代中间环节 5. 对我国航运业、保险业的发展有利 6. 船货衔接问题可以得到较好的解决，灵活性和机动性更好 | 1. 流程复杂 2. 若保险费与运费波动起伏较大，CIF需要承受更大风险 |

### 2.2.4数据处理—价格对照表

建立价格体系，同一种货物在同一条航线上制定定价体系对招标价有指导意义。在建立价格体系的时候，可以考虑诸多的因素例如航运指数、交货期、港口拥堵情况、政治因素等，都能在数学模型中得到体现，另外我们可以从这些方面定性分析外商选择FOB而不选择CIF。、

目前由于数据不足，我们还无法建立价格对照体系。之后我们会通过调查咨询收集更多的数据，明确具体的要求，建立科学合理的价格对照体系。

**2.2.4.1因素影响**

1. 航运指数（上交所）

（2）港口拥堵情况

2017年4月份，上海出现过一次严重的塞港情况，当时塞港的情况持续了一个多月的时间。港口拥堵是多重因素造成的。

直接导火索是天气不佳，上旬连续多天的浓雾天气影响了船舶靠泊。

同时，集运市场2M、OCEANAlliance和THEAlliance三大联盟正式运作，同时调整船期，额外增加了作业量，打破了原本就十分繁忙的上海港的作业平衡。

此外，期间上港集团旗下信息服务平台进行了一次数据更新，码头作业暂停了若干个小时。短期内运抵码头前沿的集装箱量大于靠港船舶的容量，形成暂落箱;暂落箱不断堆积，又造成船舶装卸不畅。

除了上述因素外，班轮公司超配舱位也一定程度引发港口拥堵的发生。班轮公司需要超额接订才能避免资源浪费，而在三大联盟新航线启动之前，班轮公司自然希望首航船舶都能满载出航，导致此次超配比例较高。

17年7月，上海港再次出现了港口拥堵的情况，多家船公司纷纷发布公告，表示船期受塞港影响将出现延误。

（3）政治因素

1）中美贸易战

2018年3月8日, 美国宣布对进口钢铁和铝分别课以25%和10%的重税, 同时暂时豁免对欧盟、阿根廷、澳大利亚、巴西、加拿大、韩国等经济体的钢铝关税至5月1日。特朗普的这一做法对我国钢铁行业来说是一个挑战。

加征直接出口税，压制间接出口量

尽管中国对美的直接出口量不大，但间接的钢材出口远超这个数字。而且特朗普一直将目光盯着钢材和铝，不仅会压制中国的钢材间接出口到美国的情况，同时也会对与钢铁相关的产品出口加征关税，从而抑制其出口。

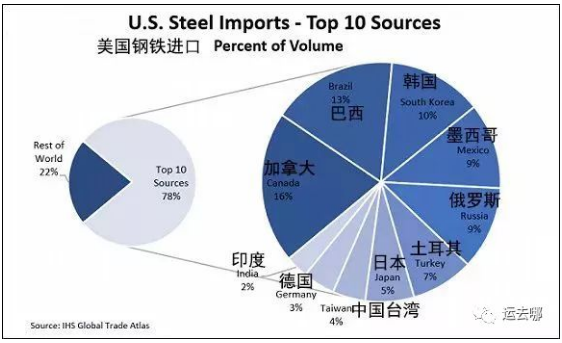


图2-4美国钢铁进口图



图2-5中国钢铁出口图

时间节点不利

特朗普这一系列举措选择的时点对我国钢铁行业不利。我国的钢铁行业供给侧实施正好两年，目前正是其边际效应出现拐点的时间点，国内钢材市场也已经进入调整阶段。

其次，美国在去年底计划大规模减税，同时实施量化宽松退出政策，美联储昨天刚刚宣布加息25个基点，年内可能还有三次加息，中国也只得被动加息。这对大宗商品和黑色系产品市场自然不利。

不确定性大大增加

由于我国钢材消费的主要市场仍然是国内市场，钢铁企业在供给侧改革后抗风险能力大幅提升，美国发起的这个贸易制裁措施对钢铁行业和钢材市场的影响是有限的和可控的，但影响也是直接的。在多种因素的综合影响下，预计我国今年的钢产量难以持续增长，将与去年持平，钢材出口量将比去年下降10%。钢材市场在未来数月将会弱势运行。

以下是根据计量模型进行的未来12个月螺纹钢价格走势预测



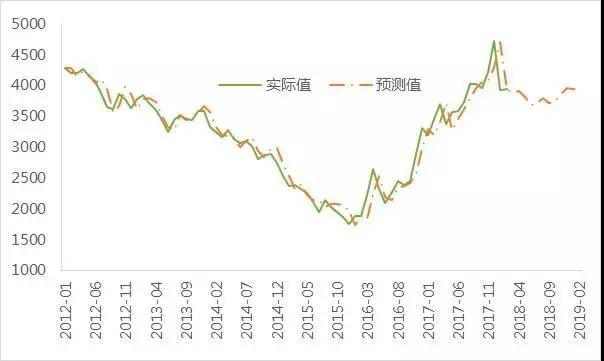


图2-6螺纹钢价格走势预测图

2）外经贸部重要通知

在出口业务的FOB合同中，有些进口商和指定的货代串通一气，采取无单提货，使中国出口企业货款两空。去年底外经贸部发出通知，要求采取严厉措施，杜绝此类现象的发生，并希望出口企业过采用CIF付款方式。

据《国际商报》的消息，外经贸部提出的措施包括：

尽量采取CIF或C&F，避免外商指定境外货代安排运输。

如外商坚持FOB条款并指定船公司和货代安排运输，可接受指定的船公司，但对货运代理的资格应进行审查，只接受经政府批准的货代。

如外商仍坚持指定境外货代，出口商应指定境外货代的提单必须委托经外经贸部批准的货运代理企业签发，并掌握货物的控制权，同时由代理签发提单的货代企业出具保函，承诺货到目的港后须凭信用证项下银行流转的正本提单放货，否则要承担无单放货的赔偿责任。

外贸公司不要轻易接受货代提单，尤其是外商指定的境外货代提单。

3）国内外供需形势

从国际看，伴随着世界经济逐步复苏，钢铁需求有所上升，而铁矿石价格的提高也推升了钢铁产品的成本。从国内情况看，我国部分基础设施建设项目的持续启动显然正在支撑对钢材的需求。此外，中国正在以前所未有的力度开展削减过剩产能工作，相关措施逐步出台，政策效应逐步显现。在削减过剩产能的同时，中国还将采取综合性措施扩大钢铁产品消费领域。

### 2.2.5定性分析与航线分析

#### 2.2.5.1定性分析出口商为什么选择FOB，而不是CIF

（1）外商角度

外商为自身利益大多要求卖家以FOB报价

1. 对险种的选择、费率、索赔等方面都有主动权；
2. 指定运输，能够掌握装运港、船期、承运人等信息，可以获得控制运输成本、获得批量运费优惠、要求运费赊账、批量进口清关等便利，方便工作开展；
3. 让买方能直观地评价各国的成本；
4. 航运公司为了市场竞争，会给出优惠的运价或者赠送航运保险，在一定程度上也降低了外商的成本；
5. 进口商在当地市场经营的产品会在中国或亚洲其它国家的数个供应商处采购，如果进口商经营的产品门类多，涉及的出口商数量会更多，进口商可以把在不同出口商采购的产品委托一家有信誉服务质量好的国际性船公司承接其海运业务，这样会减少很多运费；同样地，进口商也会把很多出口商的货柜集中投保，也会在保险公司那里得到更多的优惠。
6. 卖家角度
7. 由于国际运费上涨的风险，不利于卖方核算运费成本。估算低了卖方亏本，估算高了买方不接受，所以卖方不愿意主动使用FOB报价；

2）FOB条件较CIF流程简单，可避免拖车、驳船、报关、退税等一系列业务。还有，在FOB条件下，买方自行办理保险业务和处理索赔事宜，卖方只要及时通知买方装船资料，让买方及时办理保险业务即可。卖方认为这样更简单便利。

#### 2.2.5.2主要航线贸易术语的统计分析

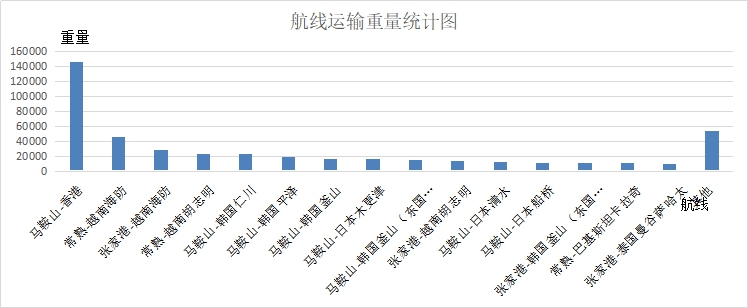
表2-7和表2-8是2017年关于各航线的合计运输重量与运输频率的统计

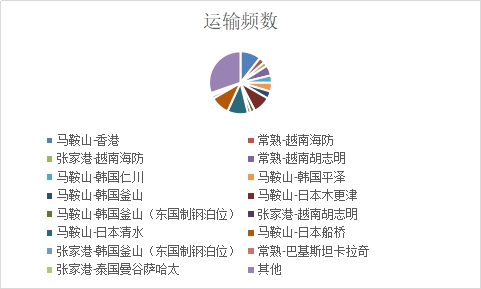
表2-7

|  |  |
| --- | --- |
| 各航线合计运输重量统计 | |
| 航线 | 重量 |
| 马鞍山-香港 | 146433.278 |
| 常熟-越南海防 | 45101.24 |
| 张家港-越南海防 | 28425.95 |
| 常熟-越南胡志明 | 23082.229 |
| 马鞍山-韩国仁川 | 22339.648 |
| 马鞍山-韩国平泽 | 19110.499 |
| 马鞍山-韩国釜山 | 16339.531 |
| 马鞍山-日本木更津 | 16182.044 |
| 马鞍山-韩国釜山（东国制钢泊位） | 15335.856 |
| 张家港-越南胡志明 | 12925.527 |
| 马鞍山-日本清水 | 12743.886 |
| 马鞍山-日本船桥 | 11246.93 |
| 张家港-韩国釜山（东国制钢泊位） | 10649.635 |
| 常熟-巴基斯坦卡拉奇 | 10299.034 |
| 张家港-泰国曼谷萨哈太 | 10195.385 |
| 其他 | 53344.359 |

表2-8

|  |  |  |
| --- | --- | --- |
| 航线运输频数统计表 | | |
| 航线 | 运输频数 | 运输频率 |
| 马鞍山-香港 | 14 | 0.101449275 |
| 马鞍山-日本清水 | 13 | 0.094202899 |
| 马鞍山-日本木更津 | 12 | 0.086956522 |
| 马鞍山-日本船桥 | 12 | 0.086956522 |
| 常熟-越南胡志明 | 7 | 0.050724638 |
| 常熟-仁川 | 7 | 0.050724638 |
| 马鞍山-韩国平泽 | 6 | 0.043478261 |
| 马鞍山-韩国仁川 | 5 | 0.036231884 |
| 马鞍山-韩国釜山 | 5 | 0.036231884 |
| 马鞍山-日本大阪 | 5 | 0.036231884 |
| 常熟-越南海防 | 4 | 0.028985507 |
| 常熟-韩国釜山 | 4 | 0.028985507 |
| 张家港-越南海防 | 3 | 0.02173913 |
| 马鞍山-韩国釜山（东国制钢泊位） | 3 | 0.02173913 |
| 上海罗泾-仁川 | 3 | 0.02173913 |
| 上海罗泾-意大利拉文纳 | 3 | 0.02173913 |
| 上海十区-马来西亚巴生 | 3 | 0.02173913 |
| 张家港-越南胡志明 | 2 | 0.014492754 |
| 张家港-韩国釜山（东国制钢泊位） | 2 | 0.014492754 |
| 常熟-比利时安特卫普 | 2 | 0.014492754 |
| 常熟-日本船桥 | 2 | 0.014492754 |
| 其他 | 21 | 0.152173913 |





选取运输重量和运输频数较大的航线重点分析

表1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-香港运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 香港 | 16.05 | 17.655 | 20 | 马鞍山 |
| 2 | 香港 | 15.9 | 17.49 |  | 马鞍山 |
| 3 | 香港 | 15.9 | 17.49 |  | 马鞍山 |
| 4 | 香港 | 15.6 | 17.16 |  | 马鞍山 |
| 5 | 香港 | 14.95 | 16.445 | 16.45 | 马鞍山 |
| 5 | 香港 | 14.8 | 16.28 | 16.95 | 马鞍山 |
| 6 | 香港 | 14.5 | 15.95 | 17 | 马鞍山 |
| 7 | 香港 | 15.7 | 17.27 |  | 马鞍山 |
| 9 | 香港 | 16 | 17.6 |  | 马鞍山 |
| 9 | 香港 | 17.5 | 19.25 |  | 马鞍山 |
| 10 | 香港 | 20.15 | 22.165 | 23 | 马鞍山 |
| 11 | 香港 | 21.5 | 23.65 |  | 马鞍山 |
| 12 | 香港 | 19.25 | 21.175 | 20 | 马鞍山 |
| 12 | 香港 | 18.5 | 20.35 |  | 马鞍山 |

1. 马鞍山-香港航线有93%选择了CIF报价

表2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 常熟-越南胡志明运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 3 | 越南胡志明 | 11 | 12.1 | 12.25 | 常熟 |
| 3 | 越南胡志明 | 10 | 11 |  | 常熟 |
| 8 | 越南胡志明 | 7 | 7.7 | 9 | 常熟 |
| 9 | 越南胡志明 | 12 | 13.2 |  | 常熟 |
| 9 | 越南胡志明 | 9.4 | 10.34 |  | 常熟 |
| 10 | 越南胡志明 | 11.3 | 12.43 |  | 常熟 |
| 12 | 越南胡志明 | 9.3 | 10.23 | 10 | 常熟 |

1. 常熟-越南胡志明有86%选择了CIF报价

表3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-韩国平泽运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 韩国平泽 | 13.7 | 15.07 |  | 马鞍山 |
| 2 | 韩国平泽 | 13.45 | 14.795 |  | 马鞍山 |
| 3 | 韩国平泽 | 12.95 | 14.245 | 14.45 | 马鞍山 |
| 8 | 韩国平泽 | 13.75 | 15.125 | 14 | 马鞍山 |
| 9 | 韩国平泽 | 13.7 | 15.07 | 15 | 马鞍山 |
| 11 | 韩国平泽 | 24.5 | 26.95 |  | 马鞍山 |

③马鞍山-韩国平泽有67%选择了CIF报价

表4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-韩国釜山 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 韩国釜山 | 12.7 | 13.97 |  | 马鞍山 |
| 2 | 韩国釜山 | 11.98 | 13.178 |  | 马鞍山 |
| 3 | 韩国釜山 | 10.45 | 11.495 | 12.75 | 马鞍山 |
| 5 | 韩国釜山 | 11.98 | 13.178 |  | 马鞍山 |
| 9 | 韩国釜山 | 13.7 | 15.07 | 15 | 马鞍山 |

④马鞍山-韩国釜山有80%选择了CIF报价

表5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 常熟-韩国釜山 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 3 | 韩国釜山 | 7.45 | 8.195 | 6.5 | 常熟 |
| 5 | 韩国釜山 | 8.5 | 9.35 | 8 | 常熟 |
| 7 | 韩国釜山 | 7.8 | 8.58 |  | 常熟 |
| 8 | 韩国釜山 | 8.2 | 9.02 |  | 常熟 |

⑤常熟-韩国釜山有50%选择了CIF报价

表6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 国内港口-韩国釜山（东国制钢泊位） | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 2 | 釜山（东国制钢） | 7.3 | 8.03 | 8.1 | 张家港 |
| 3 | 釜山（东国制钢） | 7.45 | 8.195 |  | 张家港 |
| 3 | 釜山（东国制钢） | 7.45 | 8.195 |  | 常熟 |
| 3 | 釜山（东国制钢） | 11.5 | 12.65 | 13 | 马鞍山 |
| 5 | 釜山（东国制钢） | 11.45 | 12.595 |  | 马鞍山 |
| 6 | 釜山（东国制钢） | 10.5 | 11.55 |  | 马鞍山 |

⑥国内港口-韩国釜山（东国制钢泊位）全部选择了CIF报价

表7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-日本清水运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 日本清水 | 20 | 22 | 23.65 | 马鞍山 |
| 2 | 日本清水 | 26.5 | 29.15 | 28 | 马鞍山 |
| 3 | 日本清水 | 21.8 | 23.98 | 22.45 | 马鞍山 |
| 4 | 日本清水 | 22.5 | 24.75 | 23.45 | 马鞍山 |
| 5 | 日本清水 | 17.98 | 19.778 | 20 | 马鞍山 |
| 7 | 日本清水 | 17.89 | 19.679 |  | 马鞍山 |
| 8 | 日本清水 | 22 | 24.2 | 22.45 | 马鞍山 |
| 9 | 日本清水 | 17.7 | 19.47 |  | 马鞍山 |
| 9 | 日本清水 | 19.6 | 21.56 | 20 | 马鞍山 |
| 10 | 日本清水 | 19.3 | 21.23 |  | 马鞍山 |
| 11 | 日本清水 | 30.5 | 33.55 | 31.5 | 马鞍山 |
| 12 | 日本清水 | 22.8 | 25.08 | 23 | 马鞍山 |
| 12 | 日本清水 | 18 | 19.8 |  | 马鞍山 |

⑦马鞍山-日本清水有50%选择了CIF报价

表8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-日本大阪运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 日本大阪 | 20 | 22 | 23.65 | 马鞍山 |
| 2 | 日本大阪 | 26.5 | 29.15 | 28 | 马鞍山 |
| 6 | 日本大阪 | 16.7 | 18.37 | 17 | 马鞍山 |
| 9 | 日本大阪 | 19.7 | 21.67 |  | 马鞍山 |
| 12 | 日本大阪 | 22.8 | 25.08 | 23 | 马鞍山 |

⑧马鞍山-日本大阪有40%选择了CIF报价

表9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-日本木更津运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 2 | 日本木更津 | 26.5 | 29.15 | 28 | 马鞍山 |
| 3 | 日本木更津 | 21.8 | 23.98 | 22.45 | 马鞍山 |
| 4 | 日本木更津 | 22.5 | 24.75 | 23.45 | 马鞍山 |
| 5 | 日本木更津 | 17.98 | 19.778 | 20 | 马鞍山 |
| 5 | 日本木更津 | 16.3 | 17.93 |  | 马鞍山 |
| 6 | 日本木更津 | 16.7 | 18.37 | 17 | 马鞍山 |
| 7 | 日本木更津 | 17.89 | 19.679 |  | 马鞍山 |
| 9 | 日本木更津 | 17.7 | 19.47 |  | 马鞍山 |
| 10 | 日本木更津 | 19.3 | 21.23 |  | 马鞍山 |
| 11 | 日本木更津 | 33.45 | 36.795 |  | 马鞍山 |
| 11 | 日本木更津 | 30.5 | 33.55 | 31.5 | 马鞍山 |
| 12 | 日本木更津 | 22.8 | 25.08 | 23 | 马鞍山 |

⑨马鞍山-日本木更津有50%选择了CIF报价

表10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 马鞍山-日本船桥运价对比 | | | | | |
| 月份 | 目的港 | CIF(不包含保险费) | CIF(包含保险费） | FOB | 装运港 |
| 1 | 日本船桥 | 20 | 22 | 23.65 | 马鞍山 |
| 2 | 日本船桥 | 26.5 | 29.15 | 28 | 马鞍山 |
| 3 | 日本船桥 | 21.8 | 23.98 | 22.45 | 马鞍山 |
| 4 | 日本船桥 | 22.5 | 24.75 | 23.45 | 马鞍山 |
| 5 | 日本船桥 | 17.98 | 19.778 | 20 | 马鞍山 |
| 6 | 日本船桥 | 16.7 | 18.37 | 17 | 马鞍山 |
| 6 | 日本船桥 | 8000包干 |  |  | 马鞍山 |
| 7 | 日本船桥 | 17.89 | 19.679 |  | 马鞍山 |
| 8 | 日本船桥 | 22 | 24.2 | 22.45 | 马鞍山 |
| 9 | 日本船桥 | 19.7 | 21.67 |  | 马鞍山 |
| 9 | 日本船桥 | 19.6 | 21.56 | 20 | 马鞍山 |
| 11 | 日本船桥 | 30.5 | 33.55 | 31.5 | 马鞍山 |
| 12 | 日本船桥 | 22.8 | 25.08 | 23 | 马鞍山 |

⑩马鞍山-日本船桥有33%选择了CIF报价

总结：

七月所有航线均选择CIF报价，该月可能有特殊情况

由表一、表二、表三可知，从国内港口到韩国、越南、香港的航线倾向于选择CIF报价；

对于目的港韩国釜山做具体分析，由表四、表五、表六，可看出CIF报价选择较多，对于东国制钢泊位，则全部选择了CIF报价；

由表7、表8、表9、表10可知，从马鞍山-日本的航线大部分选择FOB报价；上海十区作为装运港较易选择FOB运价，如上海十区-马来西亚巴生、上海十区-韩国浦项。

### 2.2.6 SWOT分析选择贸易术语的策略

我们通过SWOT关联度分析法对FOB和CIF贸易术语在相同的机遇和挑战下的优劣进行了分析，见下一页表2-7，表2-8。通过分析，我们为马钢在外贸交易时选择贸易术语提供了策略依据：

1）在出口业务中，尽量选择CIF贸易术语，避免外商指定境外货代安排运输。

2）如不得已采用FOB条件成交时，对于买方派船到港装货的时间应在合同中作出明确规定，以免卖方货已备好，船迟迟不到，贻误装期的事情发生。

3）对于FOB条件下，买方指定境外货代的情况应慎重考虑是否接受。作为卖方应对买方指定的货代的资质情况有一定的了解，如认为不能接受，应及时予以拒绝。

4）如外商坚持FOB条款并指定船公司和货代安排运输，可接受指定的船公司，但对货运代理的资格应进行审查，只接受经政府批准的货代。

5）即使采用信用证支付时，也应注意对托运人的规定。如果买方资信好，又有转售在途货物的要求，以买方作为托运人未尝不可。但如果不是这样，从安全起见，还是以卖方作为托运人为好。

6）提高企业的生产率水平，及时调整企业内外销的相对贸易成本对优化企业生产率结构和平衡我国内外需求水平。

## 2.3小结

本章我们通过Python语言利用KNN回归模型建立了长江运价指数数学模型，能够实时，较为准确的预测长江运价，为马钢竞标、运输、进行组织规划提供了科学的数学模型和依据。在外贸运输方面，我们定性分析了FOB和CIF的优劣，通过数据统计分析了哪些航线选择了FOB或者CIF，找到了一些共同点。通过SWOT分析为马钢外贸交易选择贸易术语提供了决策依据，有了初步建立外贸价格对照体系的想法，但由于数据不足等原因目前尚未实现。之后，我们会进一步收集数据，选取合适的方法建立外贸价格对照体系模型。





# 结构调整 提升比例

## 3.1苏锡常杭地区铁路运输比例提升方案

### 3.1.1前言

由于“长江开发原则”的提出，“马钢”在苏锡常杭地区的水路运输将会受到切实影响。为响应国家号召，同时避免码头整治对公司经营产生不利影响，也为日后取得更大的经济效益，我们提出了针对“马钢”在苏锡常杭地区的运输结构优化调整策略。

本部分首先通过对当前国家关于运输结构调整相关政策和对水运、铁运优势劣势的分析，以及对当前企业、客户状况的分析，进一步证实了提升铁运、降低水运比例的必要性。之后针对铁路比例提升存在的困难提出针对性解决方案，通过建立坐标模型，采用粒子群算法、聚类算法等大数据方法，结合传统运筹学方法进行路径规划，解决了运量如何分配、是否需要新建枢纽、新建枢纽选址等问题。同时对于近年来产生的“多批次、小批量、个性化”的钢铁物流需求，我们对钢厂内部运输布局进行了调整规划，并采用聚类方法对钢材发运中心进行选址。

接下来我们将深入调研，获取准确数据，进一步优化算法；结合当地实际情况，采用层次分析法对选址具体地点进行分析讨论；了解钢厂内部运作，完善内部结构调整方案；采用传统运筹学方法进行建模，合理规划运输路线。综合考虑效率与效益，以定量与定性相结合的方式科学合理地规划出铁路运输比例提升方案。最后将对该方案从技术、经济等方面进行可行性研究，对其投入使用后经济效果进行预测，形成可行性研究报告。

### 3.1.2背景

#### 一、政策

##### 1. 长江岸线及非法码头专项整治

为了贯彻落实党中央、国务院关于推动长江经济带发展的重大战略部署，坚持生态优先、绿色发展，共抓大保护，推进长江岸线资源高效利用，保障长江航运和生态安全，2016年10月以来，马鞍山市港航局会同市发改委、环保、水利等相关部门和单位，深入开展长江岸线及非法码头专项整治，已取得阶段性成效。马鞍山市长江干线83家和支流69家应予以取缔的非法码头全部完成拆除任务，支流和县1家正在规范提升，达到“拆除到位、清理到位、防反弹到位”，共清理出长江岸线资源约10公里，滩涂及后方土地约1000亩，一举解决了马鞍山市几十年来想解决而未能解决的问题，极大地改善了沿江生态环境。

与此同时，马鞍山市大力推进港口码头环保整治工作，建立了网格化责任体系和港口码头正面清单制度，制定了港口码头大气污染和水污染治理要求，使企业有章可循。同时开展专项整治活动，对全市所有港口码头开展全面排查，制定港口码头“一港一策”整治清单，与企业签订责任状，对标达标，完成一个，销号一个，并通过上门指导、开展培训等方式，提高企业主体责任意识，把环保行为变为企业自觉行为。在全面加强港口岸线使用管理方面，马鞍山市建立了港口岸线前置联合审核制度，实行港口岸线有偿使用，提高使用门槛，修订了《马鞍山市港口岸线办法》，出台了《马鞍山市打击非法码头建设暂行办法》，各县区结合实际建立常态长效监管机制，启动砂石集散中心建设，加强沿江监督查处，防止非法码头反弹。

对于马钢销售水运物流而言，国家对内河码头综合整治将会造成一定时期内码头资源紧张、目的港堵港严重、钢材不能及时交货等一系列问题，影响客户满意度，从而影响到公司信誉；销售物流水运发货困难、产品积压严重时，则会危及到马钢生产组织的正常进行。

##### 2.加强建设资源型产品物流集散中心

2014年9月，由国务院正式印发的《物流业发展中长期规划》提出了钢铁物流相关的三大项物流工程，并要求资源型产品物流集散中心的建设需加强，相关的配套物流通道应完善。为深入落实该规划，商务部印发《关于促进商贸物流发展的实施意见》，其中支持了业务为生产资料生产流通的企业在各个区域建设大宗生产资料物流基地与物流园区，更进一步地加快产业良性集聚。该项规划的出台，为马钢钢铁物流的发展指明了方向、提供了保障、带来了机遇。马钢应从实际角度出发，深入贯彻新发展理念和安徽省委省政府提出的“三步走”目标要求，积极融入“一带一路”，立足发展现状、结合区位特点以及物流需求，加强物流集散中心建设，完善现有钢铁物流中心，同时充分利用物流资源，以最经济的方式发展出科学高效、安全可靠的钢铁供需物流网络。

##### 3.运输结构调整方向

（一）国家关于运输结构调整相关政策

调整运输结构、提高运输效率、降低运输成本对促进经济社会发展具有十分重要的意义。当前，总体上看，我国货运结构不尽合理、交通运输综合效率较低、运输成本费用偏高问题突出。

李克强总理指出，要采取综合举措，提高大宗货物和长距离运输的铁路货运比例，特别是打通海铁联运“最后一公里”，大幅提高集装箱海铁联运比例。生态环境部提出，2018年9月底前，山东、长三角地区沿海港口煤炭集疏港运输全部改由铁路，2019年底前，京津冀及周边、长三角地区沿海港口的矿石、钢铁、焦炭等大宗货物全部改由铁路运输，禁止汽运集疏港。

2018年初以来，发改委系统研究调整运输结构、提高运输效率、降低运输成本问题，认真落实国务院第8次常务会议部署，深入推进物流领域“简政、减税、降费”工作，进一步推动降低物流成本，加快落实物流降本增效各项举措；以长江干线港口为突破口，加快推动铁水联运设施联通，大力促进铁水联运发展，大力实施打通铁路进港“最后一公里”工程。交通运输部从2018年起在全国范围实施铁路运能提升行动，加快干线铁路建设改造，提高既有铁路综合利用效率，加快铁路专用线建设，力争到2020年大宗货物年货运量在150万吨以上的工矿企业和新建物流园区接入比例达到80%以上。加快铁路市场化改革，建立灵活的运价调整机制，规范铁路专用线收费，推动铁路运输企业与大客户签订运量互保协议。

当前河北地区部分钢厂首钢、唐钢、承钢以及燕钢等企业已经陆续开始执行“公转铁”，部分钢企钢材运输成本每吨增加100元上下，矿石运输成本每吨增加40-50元，且存在亏吨、块矿粉率增加等问题。与此同时，因铁路运力不足，整体运输效率有下降。“公转铁”若全面执行，铁路运输到站后转汽运，时间成本和人力成本都会有所上升。

长三角地区，目前大部分钢厂运输仍以水运为主。因为地理优势，长三角地区河网四通八达，水运运输成本相对低廉。此项政策政治指导属性强，“马钢”作为国有钢厂，应带头执行。但对于打通铁路进港“最后一公里”工程，目前中国长三角地区铁路还不具备这样的代替作用，业内认为落地比较困难，因此目前可考虑“水转铁”。同时对于运输效率会受到影响、成本也会相应提升等问题，应结合当前“公转铁”存在的问题及自身情况综合考虑，提出完善的“水转铁”调整方案。

在中国的港口群中，长三角港口群主要包括上海、浙江和江苏两省一市的港口。从长三角港口群来看，港口腹地重叠，经济结构同质化，相互合作协调少等问题长期存在。

上海市政府发展研究中心在一份《长三角航运中心整体布局与“一带一路”国家战略衔接问题》的课题研究报告中提出，长三角港口群的内部协调性不足，呈现出合作为了竞争、竞争大于合作的阶段性特征。在政府主导式协调机制下，区域中整体合作机制尚未形成，而以上海港“长江战略”、甬舟“一体化”及苏州“三港合一”为代表的小区域合作组团参与竞争的情况显现。并且区域中已经出现“大港吃不掉、小港吃不饱”的不协调状况，各类港口资源未得到合理使用。

该课题研究报告提出，要推进长三角港口群协调发展机制改革。上海港应继续巩固“长江战略”的发展成果，持续增强对核心腹地的掌控能力，同时引领沪甬舟港口“一体化”发展，启动上海与宁波舟山港的“一体化”进程，统筹港口规范、建设与运营，构建国际转运组合枢纽。此外，更加强调以市场为主体来配置资源，弱化政府对港口发展调控的行政权力。建议上港集团可以采取更加积极的方式参与区域港口的投资运营，扩大控股港口范围、提升股权比重。在集疏运体系方面，着力开展海铁联运系统建设。以上海国际航运中心为核心重构亚欧转运体系，布局上海、宁波舟山、南京、苏州、连云港等海铁联运枢纽港，构建东北亚—长三角—中东、欧洲的国际海铁转运路径。

#### 二、社会

##### 1.铁路对降低社会物流成本的作用分析

近年来，我国物流运行仍保持了中高速增长趋势，运行效率不断提升。2017年，全国社会物流总额达到251.8万亿元，同比增长6.7%，呈现稳步增长趋势。同时，社会物流总费用也保持增长趋势，但是社会物流总费用占社会物流总额的比率呈下降趋势，由2009年的6.31%降至2017年的4.79%。此外，由于国家重视物流业降本增效，我国社会物流总费用占GDP的比率有一定的下降，从2013年18%降至2017年的14.6%，即2017年每百万元GDP所消耗的社会物流总费用为14.6万元，比上年下降1.4%。但是与发达国家（如美国8.2%），甚至一些发展中国家（如印度13%）相比，我国社会物流成本占GDP的比率仍然偏高。造成我国社会物流成本居高不下的原因有多个方面，其中铁路在降低社会物流成本方面存在一系列问题，见表3-1；铁路降低社会物流成本的作用机理见表3-2。

表3-1铁路在降低社会物流成本方面存在问题

|  |  |
| --- | --- |
| 问题 | 内容 |
| 铁路基础设施建设还需完善，铁路系统作用发挥不充分 | 随着我国中西部地区铁路建设步伐加快，跨区域快速通道基本形成，高速铁路逐步成网，城际铁路发展起步，路网规模不断扩大，保障能力明显增强。但是，我国铁路覆盖范围还需进一步扩大，西部路网稀疏，路网结构还需进一步优化。近几年，我国在铁路物流中心建设方面有了一定的进展，基本覆盖了通达铁路的重要城市，为我国发展铁路现代物流提供了良好的支撑。此外，由于设施设备和人才配置等方面的限制，现有的铁路货运场站功能不完备。铁路在路网建设、物流中心建设、货运场站的改造等方面存在的不足，使得由“点+线”构成的铁路物流网络不够畅通，铁路系统作用不能充分发挥。 |
| 多式联运与铁路供应链发展滞后 | 我国多式联运发展滞后，2016年我国多式联运占全社会货运量仅2.9%，远低于美国的40%和欧洲的35%；我国铁路集装箱运力不足，2016年铁路集装箱运量仅占铁路货运量的5%，不仅低于发达国家30%-40%的水平，也显著低于全球 20%的平均水平。此外，多式联运市场环境不完善、各枢纽站场间衔接不顺畅、运行规则不协调、装备应用水平和标准化程度低等问题突出，制约着多式联运的发展。目前，中国铁路供应链中，各个环节分割，内部资源整合能力低下；对于外部资源，铁路的整合能力不足；在增值服务方面同样发展滞后；铁路货运僵化的运作及管理体制使它难以与其它运输方式及供应链上下游企业建立起有机的联系，严重制约着铁路供应链的发展。 |
| 人力、技术等要素发展滞后 | 随着铁路物流的发展，铁路物流人才缺乏的问题越来越明显，既缺乏能制定符合我国铁路物流发展运行规则的战略型管理人才，又缺乏能确保公司科学、安全、高效运作的高级经营管理人才；既缺乏大量的能提升铁路物流仓储、包装、加工、配送各环节效率的运营管理型人才，又缺乏具备一定物流知识和一流操作经验的操作型人才。此外，铁路物流在信息化建设方面也有不足之处，虽然为了适应市场化需求，目前在铁路货运受理、仓储、运输、调度等环节都研发应用了信息系统，但由于信息系统部分功能不齐全，且各环节间数据更新不一致，使得数据不能互联、信息不能共享，（大数据物流）致使铁路物流效率提升不明显，难以实现物流全程服务。 |

表3-2铁路降低社会物流成本的作用机理

|  |  |
| --- | --- |
| 方式 | 内容 |
| 通过完善路网，完善铁路物流体系，降低系统成本 | 通过完善“点+线”的铁路物流网络，能够优化产业空间布局和城市空间布局，压缩时空成本；连接商品生产、流通、消费，提高国民经济运行效率，降低铁路物流的系统性成本。 |
| 通过优化运输结构，降低社会运输成本 | 经测算，铁路货运周转量在全社会货运周转量中的占比每增加1%，社会物流总费用与GDP的比率将降低近0.1%。说明我国目前运输结构并不合理，铁路货运市场份额占比低。  因此，在结构层面，通过在大宗散货市场提升铁路货运市场份额，发挥其运距长、运量大、速度快、费用低、连续性强、安全环保等优势，实现地区间物资流通、资源配置，利用铁路运输经济性来降低全社会运输成本 |
| 通过多式联运、供应链服务，降低物流运营成本 | 组织层面，通过开展多式联运，降低中转装卸成本。多式联运作为一种高效的货运组织运输方式，能够提高一体化运输服务水平，充分利用铁路资源，发挥其强大的影响力和组织力，使运输各环节无缝衔接，降低运输交易成本和中转装卸成本；同时，开展以铁路为核心的多式联运能够增加铁路货运量，发挥铁路骨干作用，降低运输成本、管理成本，减少能源消耗，从而降低社会物流成本。  通过发展铁路供应链，加强与上下游企业间的联系，降低供应链成本。通过与生产、商贸、金融、互联网企业等上下游企业建立起有机联系，对整条供应链上商流、物流、资金流、信息流进行集成管理，提高铁路物流服务质量与效率，能够间接降低企业的仓储、管理、资金占压、沟通等方面的成本，从而降低铁路供应链成本。 |
| 通过人力资源、技术装备的改进，提升物流效率 | 要素层面，通过提高人力资源素质，提高劳动生产率。人力资源是铁路物流系统中重要的一部分，高素质人才队伍及合理的人员结构配置，对降低社会物流成本有一定的作用。他们能够做出更好的决策，提升铁路货运服务水平；优化运输组织，减少资源浪费；提高铁路物流服务质量，提升铁路货运市场竞争力。同时，提高员工工作效率，优化人员结构能够降低人工成本，减少管理费用。通过改善作业设备，提高运输效率。改善节点的装卸作业设备，对老旧设施设备进行更新改造，提高设备专业化水平，能够减少人力成本，提高生产效率，同时能够提高节点的接卸能力，减少接取送达车辆的等待时间，加快货物周转，减少站内拥堵现象，从而降低企业运力成本、设备投入、动力成本、人员费用等，达到降低社会物流成本的作用。此外，还可更新研发新设备，降低设备购置成本、维修成本。 |

##### 2.水运转铁运对“马钢”公司影响分析

苏锡常杭地区水系发达、运价低、服务优，从而导致大部分客户都选择了通过水路发运。但水路运输本身存在着受自然条件影响大、连续性差、运输周期长、安全性较低等局限，又受到长江航道整治的影响，为避免水路受限从而对企业造成损失，调整马钢运输结构、降低水路运输比例已是大势所趋。

结合马钢物流的情况来看，铁路运输的优点如表3-3：

表3-3“马钢”铁路运输优势

|  |  |
| --- | --- |
| 铁运优势 | 内容 |
| 能够满足终端客户对产品及时交付的要求 | 在客户对交货期敏感，价格波动导致市场风险等情况下，缩短物流周期，满足产品及时交付。 |
| 保证物流安全的需要 | 在国家环保管控、重大事件或汛期时，水运物流通道受阻，但铁路物流通道畅通。 |
| 降低物流成本的需要 | 相比汽运物流而言，铁运物流能够降低物流成本。 |
| 减少货损，提高物流服务质量的需要 | 水运物流转为铁运物流，能够降低货损，提高物流服务质量。 |

此外，上海铁路局和马钢是路企战略合作关系，马钢原燃料安全保供离不开铁路局的大力支持，所以从战略合作角度，企业需支持铁路货运增量需求。但是选择采取何种运输方式的主导权在于客户，所以铁路运费下浮至接近水运价格是触发客户改走铁路运输方式的重要动因。上海铁路局基本同意给予马钢在浙江地区铁路运费下浮优惠，前提条件是马钢发往苏锡常杭地区的钢材总量要达到一定规模。

综上所述，为响应国家政策，降低社会物流成本，适应公司生产结构调整、转变物流经营管理模式以及同铁路局议价的要求，马钢发往苏锡常杭地区的水路运输比例需大幅降低，向铁路运输方向进行调整，并注意在水路运输方面促进港口集群协调发展。同时以运输比例调整为契机，进一步加强物流中心建设，促进企业供应链发展，从而节约物流成本，增加企业效益，同时提升用户体验，同用户共同发展。

### 3.1.3运能安排现状及问题分析

#### 一、客户需求分析

近年来，钢铁物流需求小微用户增多，产生了“多批次、小批量、个性化”的物流要求。在过去短缺经济条件下，钢铁产品供不应求，相应的物流方式是“大批量、小批次”。受经济增速放缓影响，钢铁产品的需求增速将放缓，需求量将保持相对稳定，而钢铁供应链中的小微用户逐渐增多。小微用户的钢铁物流需求往往分布广泛，具有一定不确定性，每次需求量较小、对于物流的时效性和个性化要求较高。目前，马钢钢铁服务较为单一，仍然依靠传统的仓储、加工、运输服务获得利润，缺乏创新的业务模式，持续盈利能力较弱。同时，铁路运输本身即存在外发装车的时效性不强、时空灵活性不高、近距离费用高、行车路线固定等缺点。因此，钢铁物流服务往往难以满足用户的时效性要求。

#### 二、企业经营状况分析

苏锡常杭地区是马钢钢材销售产品重要的集散地，2017年，从马鞍山发往苏锡常杭地区的产品总量约有150万吨，水路运输量约达到140万吨（其中，苏州约12万吨、常州约22万吨、无锡约38万吨、杭州约64万吨），公路长途运输约6万吨，铁路运输约7.5万吨，水路运输占比约91%，铁路运输占比约5%。销售的主要产品为卷钢、线螺、型材等。”公司在苏锡常杭地区的运输方式主要依靠水运，水路运输一旦受限，必将影响到整个公司的销售物流发运。

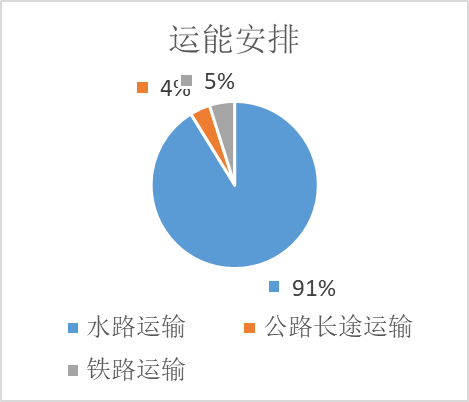
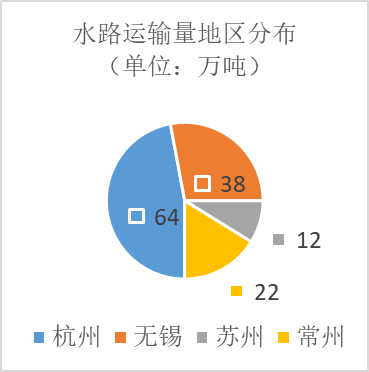
 

图3-1目前苏锡常杭地区运能安排 图3-2水路运输地区分布

（根据运输方式的不同制作的比例图） （根据地区的不同制作的水路运输的比例图）

#### 三、运能安排问题分析

表3-4铁路比例提升问题

|  |  |
| --- | --- |
| 问题 | 内容 |
| 调整范围巨大 | 2017年从马鞍山发往苏锡常杭地区的产品中铁路运输仅占比约5%，而水路运输占比高达91%。苏锡常三市仅有常州一个铁路到站，需要降低运量的水运港口却多达30个，运量高达72万吨。杭州附近铁路到站虽有萧山西、杭州北两站，但要承担杭州64万吨的水运货物依旧困难。此外，铁路运输货物种类仅有线棒材和型材。 |
| 外部制约条件 | 随着国家运输结构的调整，大力发展铁路运输，原先不符合规范的公路运输大量向铁路运输转移，而铁路方面接卸能力，货运车辆、物流枢纽节点等方面都短时间内无法满足铁路货运量大幅攀升的需求，铁路局现有的铁路站点只能完成运输、仓储、装卸等基本的物流服务功能。另外，铁路定价机制灵活度不够，价格下浮政策往往跟不上我们企业实际需求。铁路货场缺乏相关的配套加工设施、设备，其服务能力也有待提高。 |
| 自身制约条件 | 当前马钢在厂区内实施的是区域集中调度集中运输，这样能够有效地减少对机车的配置数量，提高运输效率也提高站场的使用效率。但对于冶金行业来说本身就存在发展铁路运输的劣势，生产出来的产品需要及时运出，而铁路运输存在外发装车的时效性不强、时空灵活性不高、近距离费用高、行车路线固定等缺点，难以满足发货的要求。同时，钢厂布局分散，装车点也是点多线短，下属各车站运能不平衡，一批货位数量少，运输网规模总量不足，不能满足集中发运。 |

### 3.1.4铁路运输比例提升方案

#### 一、综述

为满足铁路货运量大幅攀升的需求，弥补铁路局现有铁路站点功能的不足，我们考虑针对水运情况与钢铁物流需求，通过粒子群模型算法及聚类算法进行选址，在苏锡常杭地区布局一个或几个铁路物流枢纽，通过升级改造“传统铁路货场”的方式，打造智能化、低成本、高效率、服务于钢铁物流的铁路枢纽，实现资源和物流要素的有效配置，带动运输链条和当地产业链的发展。同时构建铁路物流配套网络，向铁路供应链两端延伸，集仓储、加工、配送、钢材市场、金融服务等产业于一体，在提供便捷服务的同时使得铁路物流综合成本最低。

对于内部铁路发运受限的问题，我们将结合环保整治的要求，根据实际需要对马钢内部物流布局做整体物流规划，构建聚类选址模型将各装车点向临近运能较大的装车点集中，从而增加一批货位数量，减少装车点，并通过增设装卸线，改造发货点来提升各货运点的铁路发运能力，并考虑建立钢材销售成品发运中心，来进一步提升铁路发运能力。

#### 二、苏锡常杭地区运输调整方案

###### （一）铁路运量分配步骤

1、准备阶段

产品：卷钢、线螺、型材

各码头位置及当前运量、现有铁路站点位置、运力、升级改造运力上限：之后获取准确数据。

2、地区选择阶段

由于各水运到港码头与客户距离相近，所以可以以码头位置及运量近似代表需求环境。接下来通过获取各水运到港码头经纬坐标、运量及常州、杭州北、萧山西三个铁路站点的经纬坐标进行建模，首先利用粒子群算法将各码头货物分配至现有铁路站点，通过分析各铁路站点运量是否均衡来判断是否需新建铁路枢纽。若不需新建，则将货运量超出阈值较多的一或两个铁路站点升级改造成为钢铁物流枢纽。若需要新建，则选取现有铁路货场平均货运量作为阈值，筛选出超出阈值范围、需求环境参数较弱的码头进行均值漂移聚类分析，忽略数据点过小的类，从而确定新建铁路物流枢纽大致区域、数量及预估运量。

3、具体地点选择阶段

对于新建铁路枢纽的选址，接下来对上一步结果附近地区进行深入调查和勘测，查阅当地的有关气象、地质、地震、水文等部门调查和研究的历史统计资料，收集供电、通信、给排水、交通运输等资料，研究运输线路以及公用管线的连接问题，收集当地有关建筑施工费用、地方税制、运输费用等各种经济资料，采取层次分析法综合考虑，在铁路沿线或现有铁路站点新增物流枢纽，从而确定最终地址。

再次使用聚类分析向已有的三个铁路站点及新建站点分配各码头，之前被忽略的数据点以及距客户较近的水运码头将依旧使用水路运输方式。根据上述因素及运量对已有铁路站点进行综合评价，选取一到两个铁路货场升级改造成为钢铁物流枢纽，从而在实现最优方案的同时使得总成本最小。

图3-3运输调整设计图

（我们方案的总体步骤流程图）

###### （二）铁路运量分配算法实现

1、算法概述

（1）改进粒子群算法简介

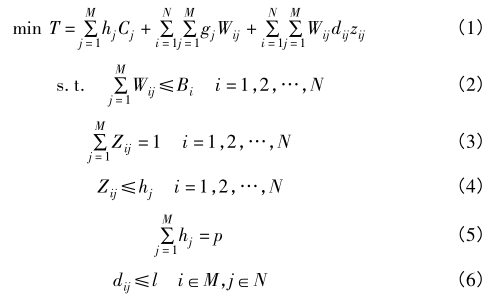
标准粒子群算法容易早熟收敛，容易陷入局部最优解。我们使用的改进算法通过引入领域均值，兼顾了粒子自身的认知思考、群体的全局最优位置和群体间的隐性知识，同时利用边界缓冲墙对越界的粒子进行缓冲，根据粒子的越界速度分情况进行处理，使算法的收敛速度和精度得到提高。

领域半径 R：在第t次迭代时粒子i的个体最优值为Pid(t)，所有粒子的全局最优值为 Pgd(t)，那么领域半径R为：



粒子群算法在寻优的过程中，粒子的位置和速度很有可能会突破其边界值，针对这种情况，采用边界缓冲墙的方法对越界粒子进行操作。

（2）选址模型建立



式(1)优化目标为总成本最小

式(2)表示分配至该站点的需求量小于等于站点规模容量

式(3)保证了每个码头将货物分配至离它最近的铁路站点

式(4)表示没有配送中心的地点不会有客户;

式(5)表示有 p个需求点被选为配送中心

式(6)表示配送中心只在限定的范围内对附近的需求点进行供应

（3）粒子群算法求解步骤

①初始化种群：确定粒子群规模、随机位置、粒子速度、惯性权重ω、加速因子c1,c2,3、缓冲墙厚度L、最大迭代次数、粒子的位置区间为［0，mj］，速度区间为［0，mj］。

②计算每个粒子的适应度值，确定个体最优值和群体最优值，计算领域均值。

③更新粒子的位置和速度。

④启动边界缓冲墙对冲出边界的粒子进行缓冲，根据粒子的实际情况对越界粒子分别进行处理。

⑤更新粒子的个体最优值、群体最优值和领域均值。

⑥判断是否满足寻优结束条件，若满足则终止寻优，输出最优结果；不满足条件则转到②。

（4）K-Means 算法简介

K-Means算法是聚类算法里面最简单的一种而且很好操作，虽然它简单但是它的效果却不可小视。正是因为它比较基础所以可以根据具体的问题加不同的权重函数，更加契合具体的问题。在本次数据模拟中，因为数据是模拟的，所以暂时没有考虑其它的权重函数，直接用了欧几里德距离。

在聚类问题中，给我们的训练样本是[clip_image004](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601448982.png)，每个[clip_image006](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601453159.png)，没有了y。

K-means算法是将样本聚类成k个簇（cluster），具体算法描述如下：

①随机选取k个聚类质心点（cluster centroids）为[clip_image008[6]](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601454064.png)。

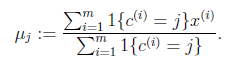
②重复下面过程直到收敛

{

对于每一个样例i，计算其应该属于的类

[clip_image009](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601464654.png)

对于每一个类j，重新计算该类的质心

[](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601468308.png)

}

（5）均值漂移聚类简介

均值漂移聚类是基于滑动窗口的算法，来找到数据点的密集区域。这是一个基于质心的算法，通过将中心点的候选点更新为滑动窗口内点的均值来完成，来定位每个组/类的中心点。然后对这些候选窗口进行相似窗口进行去除，最终形成中心点集及相应的分组。不同于K-Means算法，基于密度的算法相比于K-Means受均值影响较小，并且均值漂移聚类算法不需要我们知道有多少类/组，有利于准确判断是否需新建铁路物流枢纽，同时准确确定枢纽位置、预估运量。在获得真实数据后，我们会使用均值漂移算法，其实它也可以说是K均值聚类算法的变体。考虑了范围半径，算是距离权重的另一种表现方式。但是范围半径的选取是十分重要的，这不仅要从数据上分析还要综合考虑现实生活。这就需要调研很真实数据的支持。

（6）均值漂移聚类求解步骤：

① 确定滑动窗口半径r，以随机选取的中心点C半径为r的圆形滑动窗口开始滑动。均值漂移类似一种爬山算法，在每一次迭代中向密度更高的区域移动，直到收敛。

② 每一次滑动到新的区域，计算滑动窗口内的均值来作为中心点，滑动窗口内的点的数量为窗口内的密度。在每一次移动中，窗口会想密度更高的区域移动。

③ 移动窗口，计算窗口内的中心点以及窗口内的密度，知道没有方向在窗口内可以容纳更多的点，即一直移动到圆内密度不再增加为止。

④ 步骤一到三会产生很多个滑动窗口，当多个滑动窗口重叠时，保留包含最多点的窗口，然后根据数据点所在的滑动窗口进行聚类。

2、数据采集与描述

根据水运总运量140万吨和现实码头运量的考虑，我们模拟了现实的数据以及模拟了铁路站点改造后的运量上限（如果和现实差距较大，请见谅，初稿没找到数据）

现有铁路站点升级改造运力上限和现有码头承运量：

图3-4当前铁路运量模拟

（先根据总水路运量的80%确定总和，然后根据实际情况分配各站点的实际运量，得出上图的结果）

图3-5码头吞吐量数据模拟

（各个码头的吞吐量，根据查到的资料和数据的结合实际情况的数据模拟出的数据）

3、实现结果

（1）运用粒子群算法实现运量的初步分配

分配运量结果如图3-6所示：

图3-6粒子群算法运量分配图

（运用粒子群算法为各个站点分配运量，前面是站点可以承受的运量，后面是把码头运量通过粒子群算法模型分配给各个站点。粒子群模型是先随机构建一个种群，每个个体是一个解，然后通过修改个体来得到最优的解）

可知需要新建铁路站点，杭州北和常州-石油分公司专用这两个站点承受运量过多，需要新建铁路站点来承受多出来的运量。

（2）由上可知四个站点明显不够，需要新建站点。为此我们还利用了K-Means算法把码头分成了五类。由欧几里德距离作为最小化的依据。初步结果如下图

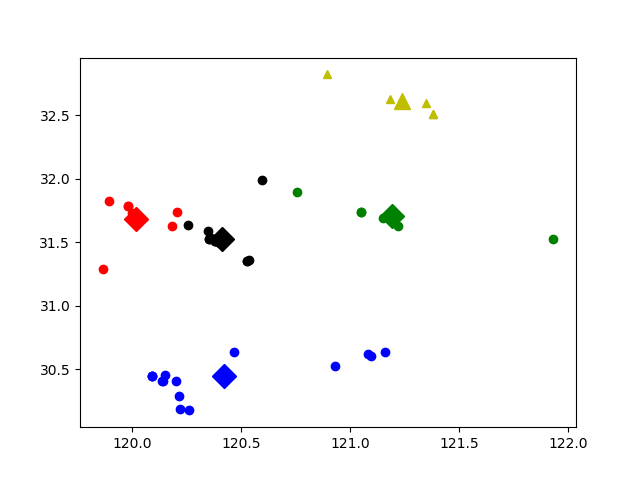


图3-7 码头聚类结果图

（通过K均值算法聚类，把码头根据距离聚类成五个。展示出了码头能根据距离的远近分成五类，站点的新建和升级可以有一个依据。当然也能调整聚类的类数）

再根据已有的站点分析和码头运量分析哪些码头该分配到新站点

（3）由数据分析可得出需要找到一个新的铁路站点来分配以下码头的运量：

苏州开元物流码头

常熟亨通港务有限公司

杭州仁和港码头

浙江佳供码头

杭州武林头申江码头

乍浦农资码头

常州武进钢材市场码头

无锡新中宝库2＃港119航段

无锡海德码头

无锡新安锚地港

（4）码头大多集中在苏州和无锡地区，所以我们决定新建站点在苏州附近

因为需要分配站点较少而且集中，所以本次模拟直接采用K-Means得出合理的选址范围，再加上适当的环境考量和分析。

K-Means结果如图3-7所示：

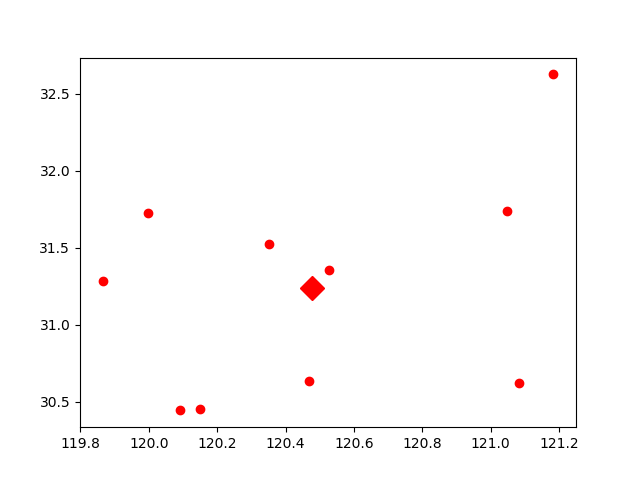


图3-8K-Means模拟选址结果

（根据实际情况分析得到的需要分配到另外站点的聚类中心）

再根据实际地理位置以及实际的情况考虑得出最终结果如下

①铁路沿线，不需要再新修铁路

②靠近主干道，靠近码头



图 3-9 站点在地图上的位置截图

当新建铁路站点之后再运用粒子群算法分配运量，结果如图3-8所示：（单位万吨）

图3-10新建站点后运量分配图

各码头位置和铁路站点位置以及新建铁路站点位置，如图3-9所示：

图3-11码头和铁路站点分布图

4、结论

由以上数据处理的结果可以知道，因为铁路站点的稀少，在加上如此多的码头需要承运，以及铁路站点分布的不均匀，苏州和无锡没有铁路站点，只有分配到常州的站点。所以我们综合数据和实际情况考虑，杭州北的站点改造升级至承运量30万吨。对于常州的站点，因为要分担苏锡常地区的运量，常州铁路站点升级至30万吨。再在苏州望亭铁路沿线新建一个站点，或将望亭东站升级改造用于钢材运输，因为需要承担运量较多，所以承运上限需达到40万吨。

##### （三）具体地点选择

1、铁路枢纽选址影响因素分析：

经营环境：铁路沿线；靠近主干道、码头、物流园等

自然环境：远离居民区，常年风向（东南偏东）背离居民区

建厂条件：地形平坦开阔，拆迁量小

城市规划：通过对江苏已建成的京沪高速铁路、沪宁城际铁路等沿线枢纽周边开发现状和规划情况的梳理，主要存在五个方面问题。

一是枢纽与城市距离过大。枢纽选址远离城市原有中心，枢纽周边发展动力不足，要素难以集聚，同时也导致基础设施建设成本过高，城市能级难以支撑；

二是枢纽周边用地规模过大。该问题在中小城市较为突出，用地开发规模缺乏与城市规模、枢纽规模的协调；

三是枢纽功能定位与城市能级不匹配。发展定位过高，围绕枢纽建设新城、新区，与城市能级不匹配；功能定位雷同，缺乏与城市自身产业的协调；

四是枢纽周边产业与城市发展不协调。产业定位与城市发展阶段适应性不足。部分小规模枢纽提出重点发展总部经济、会议商务等类型，依据目前的发展条件短期内难以实现。业态引导与城市功能提升契合性不强，如部分枢纽提出产业发展与城市原有中心区产业雷同，难以体现枢纽周边的发展优势。

五是枢纽周边缺乏特色，枢纽地区形象单一，掩盖城市个性，枢纽地区建设缺乏与城市文化、地域特征的呼应。

（来源：江苏城乡规划网站）

2、层次分析法确定最终地点

##### （四）路径规划

确定铁路枢纽具体位置后，获取当前铁路状况相关信息，根据各站点距离及预估运量采用节约法合理规划铁路运输线路，并结合周边环境、客户需求构建铁路物流配套网络。

#### 三、内部铁路优化

近年来，钢铁物流需求小微用户增多，产生了“多批次、小批量、个性化”的物流要求。在过去短缺经济条件下，钢铁产品供不应求，相应的物流方式是“大批量、小批次”。受经济增速放缓影响，钢铁产品的需求增速将放缓，需求量将保持相对稳定，而钢铁供应链中的小微用户逐渐增多。小微用户的钢铁物流需求往往分布广泛，具有一定不确定性，每次需求量较小、对于物流的时效性和个性化要求较高。目前，多数钢铁物流企业服务较为单一，仍然依靠传统的仓储、加工、运输服务获得利润，缺乏创新的业务模式，持续盈利能力较弱。同时，由于钢铁物流企业分布具有一定的区域聚集特点，并且，不同区域之间的钢铁物流企业缺乏相应的协作，因此，钢铁物流服务往往难以满足用户的时效性要求。

对于内部铁路存在的问题，应结合环保整治的要求，根据实际需要对马钢内部物流布局做整体物流规划，优先满足大批大量用户需求，并对大批量需求较为集中的几个发货点进行扩建，从而增加一批货位数量，减少装车点，并通过增设装卸线，改造发货点来提升各中心货运点的铁路发运能力。同时，其余较小装车点可用于承担中小批量、对时效性要求较高的客户需求。此外可考虑建立钢材销售成品发运中心，来进一步提升铁路发运能力。对于钢材销售成品发运中心的选址问题，可根据各钢厂位置、产量盈余情况，使用K-Means聚类在铁路沿线进行发运中心选址。

### 3.1.5可行性研究

此部分将在获取到准确数据、方案确定之后再进行详细分析，编制财务报表。

#### 一、铁路枢纽建设成本预算

建设成本包括苏锡常杭地区铁路站点新建、扩建成本以及厂内布局优化改造成本。

1、生产规模

该项目建成后拟承担苏锡常杭地区马钢钢材及其它大宗货物运输。杭州北站升级至承运量30万吨，常州站升级至30万吨，苏州望亭新建站点承运量40万吨。

2、实施进度

该项目拟二年建成，第三年投入运营，年均承运量达设计承载能力的80%。

3、建设投资估算

4、流动资金估算

5、投资使用计划与资金来源

建设投资分年使用；流动资金从投产第一年开始按运载负荷进行安排。

#### 二、运营成本估算

按照总成本观念,物流成本分析应包括满足物流需要所必需的全部成本。铁路运输总成本应包括运输成本、库存成本、装卸成本和管理成本（也称交易成本）。

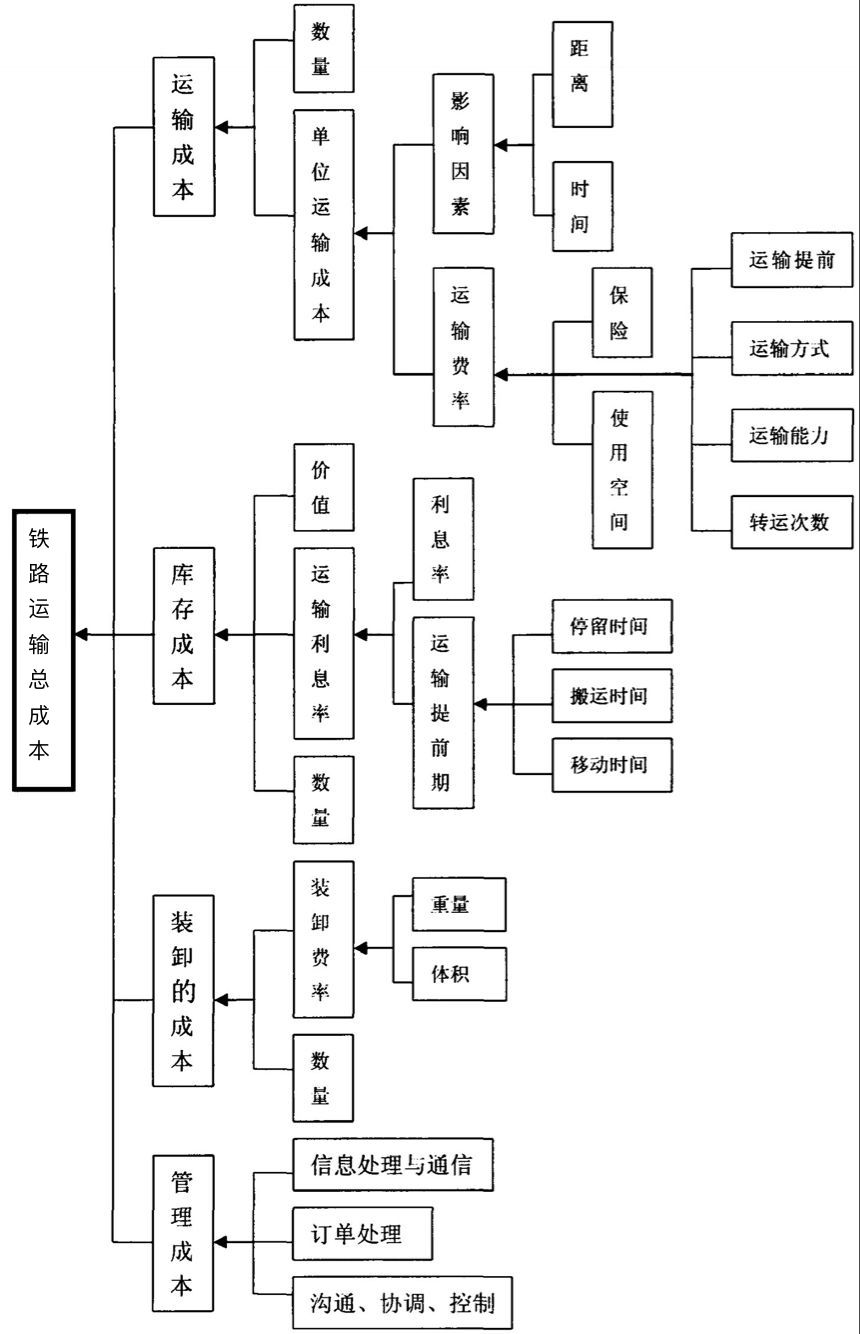


图3-12总成本分析图

#### 三、运营收入及利润估算

根据水运运价大致确定铁运定价，相关税收根据实际进行测算。

#### 四、评价参数

计算财务净现值、内部收益率、静态和动态投资回收期等财务评价指标，进行盈利、清偿能力和风险分析。

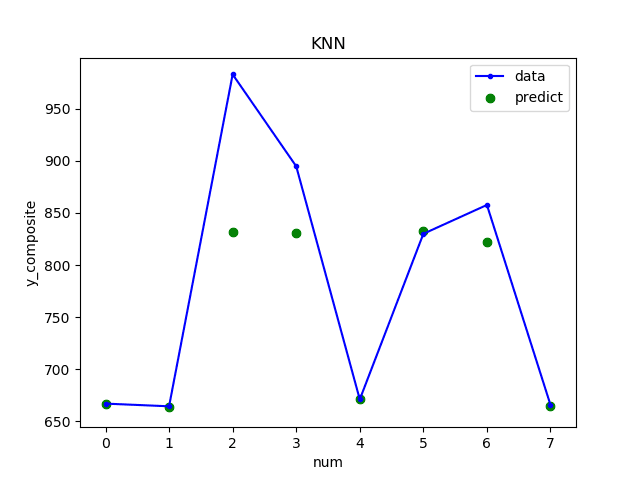
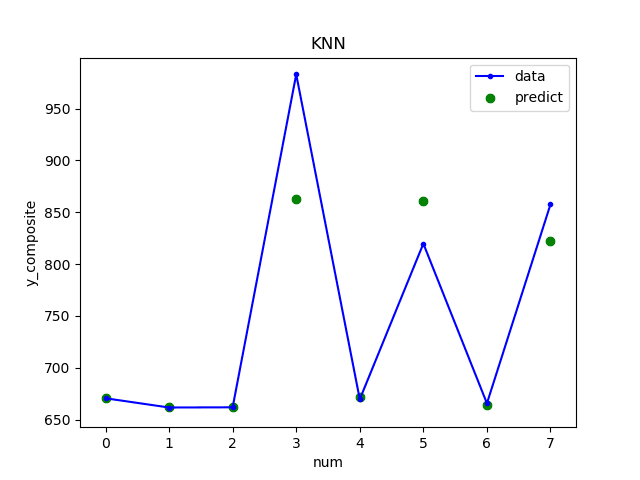
## 3.2小结

我国社会物流成本占GDP的比率仍居高不下，铁路运输的发展改革已然成为必然趋势。我们通过数据分析和模拟建立数学模型对铁路物流枢纽进行选址，从而提高苏锡常杭地区的铁路运输比例，调整企业的运输结构。铁路运输比例的提升，于企业而言，不仅避免了短期内整治可能造成的不利影响，更是对马钢服务水平的提升起到至关重要的作用，大大改善了客户的体验；于国家而言，减轻了内河运输所造成的污染与负担，同时也促进了国家路网及大宗货物运输的发展。与此同时，还可同步完善供应链，加强建设资源型产品物流集散中心，从而进一步促进钢铁行业的繁荣，为企业创造出更大的效益。

由于时间和数据等原因，我们未能完成苏锡常杭地区铁路运输比例提升方案的可行性分析报告，之后我们会进一步完善可行性分析报告。

# 总结

第二章长江运价分析中，我们通过数据分析选取了影响长江运价的重要因素，分别是海运综合指数，煤炭指数，粮食指数，金属矿石指数，原油指数，成品油指数，然后用Python和C++语言利用KNN回归模型最终训练出了了长江运价指数模型（详细请见2.1长江运价分析），以下是我们拟合的结果：



长江运价模型拟合结果图

我们可以通过该实时预测模型，较为准确的预测长江运价指数，为马钢竞标、运输、进行组织规划提供了科学的数学模型和依据。

在外贸运输方面，我们定性分析了FOB和CIF的优劣，通过数据统计分析了哪些航线选择了FOB或者CIF，找到了一些共同点。通过SWOT分析为马钢外贸交易选择贸易术语提供了决策依据，有了初步建立外贸价格对照体系的想法，但由于数据不足等原因目前尚未实现。之后，我们会进一步收集数据，选取合适的方法建立外贸价格对照体系模型。

第三章中，我们通过数据分析和模拟建立数学模型对铁路物流枢纽进行选址，从而提高苏锡常杭地区的铁路运输比例，调整企业的运输结构。铁路运输比例的提升，于企业而言，不仅避免了短期内整治可能造成的不利影响，更是对马钢服务水平的提升起到至关重要的作用，大大改善了客户的体验；于国家而言，减轻了内河运输所造成的污染与负担，同时也促进了国家路网及大宗货物运输的发展。与此同时，还可同步完善供应链，加强建设资源型产品物流集散中心，从而进一步促进钢铁行业的繁荣，为企业创造出更大的效益。

由于时间和数据等原因，我们未能完成苏锡常杭地区铁路运输比例提升方案的可行性分析报告，之后我们会进一步完善可行性分析报告。

综上所述，我们还有一些想法和目标没有实现，接下来我们会朝着这些方向努力，同时我们希望能把多式联运融入到我们的方案中，目前由于数据资料有限，我们的方案还需要进一步优化。

# 参考文献

[1]刘晴,龙小娟.FOB盛行之谜:贸易成本与异质性企业的贸易术语选择[J].国际商务研究,2014,35(03):16-26+37.

[2]胡伟,徐福缘,台德艺,马庆国.基于改进粒子群算法的物流配送中心选址策略[J].计算机应用研究,2012,29(12):4489-4491.

[3]樊毓卿. 供需视角下钢铁物流资源整合模式及运作机制研究[D].北京交通大学,2018.

[4]陆东福.长三角地区国际竞争力与可持续发展中的铁路建设[J].中国铁路,2007(12):1-7.

[5]王溶花,鲍平平.我国进出口业务中选择FOB与CIF术语的风险及防范[J].经济视角(下),2009(09):55-57.

[6]焦漩,孟宪军.出口企业选用FOB与CIF贸易术语的风险比较[J].商场现代化,2015(10):4-5.

[7]张文敬.中小出口企业在实践中对FOB与FCA的术语选择问题[J].对外经贸实务,2018(02):61-63.

# 附录

梯度下降处理单变量（C语言实现）

KNN回归模型（Python实现 运用了sklearn库）

改版粒子群算法（Python实现）

K-Means算法（Python实现）

梯度下降处理单变量（C语言实现）

#include <cstdio>

#include <cstdlib>

using namespace std;

int n = 0;

double alpha = 0.0001;

double a = 1, b = 0;

double x[100], y[100];

double min\_a,min\_b,min;

double sigma\_x() {

double tmp = 0;

for (int i = 1; i <= n; i++)tmp += x[i];

return tmp;

}

double sigma\_y() {

double tmp = 0;

for (int i = 1; i <= n; i++)tmp += y[i];

return tmp;

}

double sigma\_xy() {

double tmp = 0;

for (int i = 1; i <= n; i++)tmp += x[i] \* y[i];

return tmp;

}

double sigma\_xx() {

double tmp = 0;

for (int i = 1; i <= n; i++)tmp += x[i] \* x[i];

return tmp;

}

double sigma\_yy() {

double tmp = 0;

for (int i = 1; i <= n; i++)tmp += y[i] \* y[i];

return tmp;

}

double calc\_da() {

double tmp = 2 \* b\*sigma\_x() + 2 \* a + sigma\_xx() - 2 \* sigma\_xy();

return tmp / n;

}

double calc\_db() {

double tmp = 2 \* n\*b + 2 \* a\*sigma\_x() - 2 \* sigma\_y();

return tmp / n;

}

double calc(double a, double b) {

double tmp = n \* b\*b + 2 \* a\*b\*sigma\_x() + a \* a\*sigma\_xx() - 2 \* a\*sigma\_xy() - 2 \* b\*sigma\_y() + sigma\_yy();

return tmp / n;

}

int main()

{

freopen("data.in", "r", stdin);

freopen("data.out","w",stdout);

double x\_max=0,y\_max=0;

while (scanf("%lf%lf",&x[n + 1], &y[n + 1]) != -1) {

if (x[n+1]>x\_max)x\_max=x[n+1];

if (y[n+1]>y\_max)y\_max=y[n+1];

n++;

}

for (int i=1;i<=n;i++){

x[i]=x[i]/x\_max;

y[i]=y[i]/y\_max;

}

//printf("%d\n", n);

//for (int i = 1; i <= n; i++)printf("%.2f %.2f\n", x[i], y[i]);

while (true) {

double da = calc\_da();

double db = calc\_db();

double tmp1 = calc(a, b);

double tmp2 = calc(a - alpha \* da, b - alpha \* db);

if (tmp2 > tmp1) {

alpha /= 2;

}

else {

a = a - alpha \* da;

b = b - alpha \* db;

}

if (alpha < 0.0000000000000001)break;

//printf("%.2f %.2f %.2f %.2f\n",da,db,tmp1,tmp2);

}

printf("%.5f %.5f %.5f\n", a, b, calc(a, b));

min=calc(a,b);

min\_a=a;

min\_b=b;

// double tmp=0;

// for (int i=1;i<=n;i++)

// tmp+=(a\*x[i]+b-y[i])\*(a\*x[i]+b-y[i]);

// printf("%.2f\n",tmp/n);

for (int i=1;i<=500;i++){

a=rand()%100\*1.0/10;

b=rand()%20-10;

while (true) {

double da = calc\_da();

double db = calc\_db();

double tmp1 = calc(a, b);

double tmp2 = calc(a - alpha \* da, b - alpha \* db);

if (tmp2 > tmp1) {

alpha /= 2;

}

else {

a = a - alpha \* da;

b = b - alpha \* db;

}

if (alpha < 0.0000000000000001)break;

//printf("%.2f %.2f %.2f %.2f\n",da,db,tmp1,tmp2);

}

if (calc(a,b)<min) {

min=calc(a,b);

min\_a=a;

min\_b=b;

}

}

printf("%.5f %.5f %.5f\n", min\_a, min\_b, calc(min\_a,min\_b));

return 0;

}

KNN回归模型（Python实现 运用了sklearn库）

（1）初步模型代码

import numpy as np

import random

import matplotlib.pyplot as plt

from sklearn import neighbors

import math

M = 37 # 数据个数

N = 3 # 属性个数

X = np.zeros((M, N)) # 数据集

Y = np.zeros((M, 1)) # 结果（长江航运综合指数）

def load\_data(): # 加载数据

with open("afterData/new\_oil\_price.txt", 'r') as f: # 加载油价

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][0] = float(lines[i].replace("\n", ""))

with open("afterData/sandstone\_price.txt", 'r') as f: # 加载砂石运价

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][1] = float(lines[i].replace("\n", ""))

with open("afterData/sea\_price.txt", 'r') as f: # 加载沿海运价综合指数

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][2] = float(lines[i].replace("\n", ""))

with open("afterData/composite\_price.txt", 'r') as f: # 加载长江综合航运运价指数

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

Y[i-1][0] = float(lines[i].replace("\n", ""))

def rescale(x\_data, y\_data, scale): # 把数据缩放 返回数据集 scale为缩放因子的列表

nx\_data = x\_data.copy()

for i in range(np.shape(x\_data)[0]):

nx\_data[i] = [x\_data[i][n]\*scale[n] for n in range(np.shape(x\_data)[1])]

return nx\_data, y\_data

def divide\_data(x\_data, y\_data, n): # 把数据划分成训练集和验证集 使用留出法 分层抽样 季节分层

num = np.shape(x\_data)[0]//12

num\_test = int(n\*np.shape(x\_data)[0]) # 必须是四的整数倍

x\_test = np.zeros((num\_test, N))

y\_test = np.zeros((num\_test, 1))

x\_train = np.zeros((M-num\_test, N))

y\_train = np.zeros((M-num\_test, 1))

index = [m for m in range(M)]

count = 0

for i in range(4):

for j in range(num\_test//4):

while True:

offset = random.randint(0, num-1)\*12+i\*3+random.randint(1, 3)

if offset in index:

break

x\_test[count] = x\_data[offset]

y\_test[count] = y\_data[offset]

index.remove(offset)

count += 1

for i, j in enumerate(index):

x\_train[i] = x\_data[j]

y\_train[i] = y\_data[j]

return x\_test, y\_test, x\_train, y\_train

def train(x\_train, y\_train, k, weight="uniform"):

knn = neighbors.KNeighborsRegressor(k, weights=weight)

model = knn.fit(x\_train, y\_train)

return model

def test\_model(x\_train, y\_train, k, x\_test, y\_test, weight="distance"):

knn = neighbors.KNeighborsRegressor(k, weights=weight)

guess = knn.fit(x\_train, y\_train).predict(x\_test)

error = ((guess-y\_test)\*\*2).sum()

# print(np.shape(x\_test)[0])

return error/np.shape(x\_test)[0]

def cost(x\_data, y\_data, k, trials=100, n=0.12):

error = 0.0

for i in range(trials):

x\_train, y\_train, x\_test, y\_test = divide\_data(x\_data, y\_data, n)

# model = train(x\_train, y\_train, k)

error += test\_model(x\_train, y\_train, k, x\_test, y\_test)

return error/trials

def annealing\_optimize(domain, costf, T=10000.0, cool=0.95, step=1):

vec = [float(random.randint(domain[i][0], domain[i][1])) for i in range(len(domain))]

best = 999999999999

best\_vec = vec

while T > 0.1:

i = random.randint(0, len(domain)-1)

dir = random.randint(-step, step)

vecb = vec[:]

if i == 3:

vecb[i] += dir

else:

vecb[i] += dir\*0.3

if vecb[i] < domain[i][0]:

vecb[i] = domain[i][0]

elif vecb[i] > domain[i][1]:

vecb[i] = domain[i][1]

ea = costf(vec)

eb = costf(vecb)

if eb < ea or random.random() < math.pow(math.e, -(eb-ea)/T):

vec = vecb

T \*= cool

print(vec)

a = costf(vec)

if a < best:

best = a

best\_vec = vec

print(a)

return vec, best, best\_vec

def create\_cost(X, Y, n=0.12):

def costf(vec):

X\_new, Y\_new = rescale(X, Y, vec[:3])

return cost(X\_new, Y\_new, int(vec[3]), n=n)

return costf

def main():

load\_data()

# print(X)

# print(Y)

weight\_domain = [(0, 5), (10, 20), (5, 15), (2, 5)]

costf = create\_cost(X, Y, n=0.33)

result, best, best\_vec = annealing\_optimize(weight\_domain, costf, T=1000000.0, cool=0.99)

print(result)

print(costf(result))

print(best\_vec)

print(best)

if \_\_name\_\_ == '\_\_main\_\_':

main()

# plt.subplot(2, 1, 1)

# plt.scatter(X\_test[:, :1], X\_test[:, 1:], c='y', label='data')

# plt.scatter(X\_train[:, :1], X\_train[:, 1:], c='g', label='data')

# plt.axis('tight')

# plt.legend()

#

# plt.tight\_layout()

# plt.show()

1. 加上数据展示功能

import numpy as np

import random

import matplotlib.pyplot as plt

from sklearn import neighbors

import math

"""

[2.4000000000000004, 11.500000000000004, 10.199999999999989, 2]

2193.2733507773955

"""

M = 37 # 数据个数

N = 3 # 属性个数

X = np.zeros((M, N)) # 数据集

Y = np.zeros((M, 1)) # 结果（长江航运综合指数）

def load\_data(): # 加载数据

with open("afterData/new\_oil\_price.txt", 'r') as f: # 加载油价

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][0] = float(lines[i].replace("\n", ""))

with open("afterData/sandstone\_price.txt", 'r') as f: # 加载砂石运价

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][1] = float(lines[i].replace("\n", ""))

with open("afterData/sea\_price.txt", 'r') as f: # 加载沿海运价综合指数

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

X[i-1][2] = float(lines[i].replace("\n", ""))

with open("afterData/composite\_price.txt", 'r') as f: # 加载长江综合航运运价指数

lines = f.readlines()

# print(len(lines))

for i in range(1, len(lines)):

Y[i-1][0] = float(lines[i].replace("\n", ""))

def make\_test(x\_data, y\_data):

x\_test = np.zeros((M, N))

y\_test = np.zeros((M, 1))

index = [i for i in range(M)]

for i in range(M):

m = random.choice(index)

x\_test[i] = x\_data[m]

y\_test[i] = y\_data[m]

return x\_test, y\_test

def divide\_data(x\_data, y\_data, n): # 把数据划分成训练集和验证集 使用留出法 分层抽样 季节分层

num = np.shape(x\_data)[0]//12

num\_test = int(n\*np.shape(x\_data)[0]) # 必须是四的整数倍

x\_test = np.zeros((num\_test, N))

y\_test = np.zeros((num\_test, 1))

x\_train = np.zeros((M-num\_test, N))

y\_train = np.zeros((M-num\_test, 1))

index = [m for m in range(M)]

count = 0

for i in range(4):

for j in range(num\_test//4):

while True:

offset = random.randint(0, num-1)\*12+i\*3+random.randint(1, 3)

if offset in index:

break

x\_test[count] = x\_data[offset]

y\_test[count] = y\_data[offset]

index.remove(offset)

count += 1

for i, j in enumerate(index):

x\_train[i] = x\_data[j]

y\_train[i] = y\_data[j]

return x\_test, y\_test, x\_train, y\_train

def rescale(x\_data, y\_data, scale): # 把数据缩放 返回数据集 scale为缩放因子的列表

nx\_data = x\_data.copy()

for i in range(np.shape(x\_data)[0]):

nx\_data[i] = [x\_data[i][n]\*scale[n] for n in range(np.shape(x\_data)[1])]

return nx\_data, y\_data

def main():

load\_data()

# scale = [2.4, 11.5, 10.2]

scale = [10, 5, 5]

x\_data, y\_data = rescale(X, Y, scale)

# x\_test, y\_test = make\_test(x\_data, y\_data)

x\_test, y\_test, x\_train, y\_train = divide\_data(x\_data, y\_data, n=0.22)

knn = neighbors.KNeighborsRegressor(n\_neighbors=2, weights="distance")

y = knn.fit(x\_train, y\_train).predict(x\_test)

x = np.zeros((8, 1))

for i in range(8):

x[i] = i

plt.subplot(1, 1, 1)

plt.plot(x, y\_test, '.-', c='b', label='data')

plt.scatter(x, y, c='g', label='predict')

plt.axis('tight')

plt.legend()

plt.title("KNN")

plt.xlabel("num")

plt.ylabel("y\_composite")

# plt.subplot(2, 1, 2)

# plt.plot(x, y, '.-', c='g', label='predict')

# plt.axis('tight')

# plt.legend()

# plt.title("KNN")

# plt.xlabel("num")

# plt.ylabel("y\_predict")

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

main()

改版粒子群算法（Python实现）

import random

import numpy

import math

class Bird(object):

"""

speed:速度 一个n\*1的矩阵

position:位置 n\*1的矩阵

fit:适应度 优化目标函数的值

lbestposition:经历的最佳位置

lbestfit:经历的最佳的适应度值

"""

def \_\_init\_\_(self, speed, position, fit, lBestPosition, lBestFit):

self.speed = speed

self.position = position

self.fit = fit

self.lBestFit = lBestFit

self.lBestPosition = lBestPosition

class PSO(object):

"""

fitFunc:适应度函数

birdNum:种群规模

w:惯性权重

c1,c2:个体学习因子，社会学习因子

solutionSpace:解空间，列表类型：[最小值，最大值]

"""

def \_\_init\_\_(self, fitFunc, birdNum, w, c1, c2, solutionSpace):

self.fitFunc = fitFunc

self.w = w

self.c1 = c1

self.c2 = c2

self.birds, self.best = self.initbirds(birdNum, solutionSpace)

def initbirds(self, size, solutionSpace):

birds = []

#初始化种群

for i in range(size):

position = numpy.array([random.randint(solutionSpace[0], solutionSpace[1]) for i in range(43)])

speed = numpy.zeros(43)

fit = self.fitFunc(position)

birds.append(Bird(speed, position, fit, position, fit))

best = birds[0]

#获得初始化里面最优的种群适应值

for bird in birds:

if bird.fit > best.fit:

best = bird

return birds, best

def updateBirds(self):

for bird in self.birds:

# 更新速度

change = self.w \* bird.speed

bird.speed = change + self.c1 \* random.random() \* (

bird.lBestPosition - bird.position) + self.c2 \* random.random() \* (

self.best.position - bird.position)

# 更新位置

# print(bird.speed.shape())

# print(bird.speed)

# print(bird.position)

bird.position = (bird.position + bird.speed).astype(numpy.int)

for i in range(43):

if bird.position[i] < 0:

bird.position[i] = 1

elif bird.position[i] > 3:

bird.position[i] = 3

# 跟新适应度

bird.fit = self.fitFunc(bird.position)

# 查看是否需要更新经验最优

if bird.fit > bird.lBestFit:

bird.lBestFit = bird.fit

bird.lBestPosition = bird.position

def solve(self, maxIter):

# 只考虑了最大迭代次数，如需考虑阈值，添加判断语句就好

for i in range(maxIter):

# 更新粒子

self.updateBirds()

for bird in self.birds:

# 查看是否需要更新全局最优

if bird.fit > self.best.fit:

self.best = bird

return self.best.position

def main():

dlist = [ #码头的地理位置

[120.534715, 31.361962],

[120.526348, 31.354009],

[120.526348, 31.354009],

[120.756484, 31.896384],

[120.593221, 31.991697],

[121.150376, 31.688374],

[121.217258, 31.628585],

[121.048025, 31.739041],

[120.138984, 30.404504],

[121.048025, 31.739041],

[120.140872, 30.406769],

[120.150184, 30.453025],

[120.467688, 30.636555],

[120.221546, 30.185337],

[120.092182, 30.443046],

[120.092182, 30.443046],

[120.092182, 30.443046],

[120.202197, 30.409878],

[121.157653, 30.639385],

[121.082961, 30.622273],

[121.096177, 30.604405],

[120.931349, 30.523157],

[121.931349, 31.523157],

[119.894896, 31.820257],

[120.894896, 32.820257],

[119.998021, 31.725647],

[120.403347, 31.489619],

[120.351, 31.526],

[120.256718, 31.636897],

[120.351, 31.526],

[120.36088, 31.534352],

[120.403347, 31.489619],

[120.381622, 31.508581],

[120.379141, 31.51168],

[121.379141, 32.51168],

[120.379141, 31.51168],

[121.379141, 32.51168],

[119.864515, 31.285693],

[120.207584, 31.734799],

[120.183873, 31.626063],

[121.183873, 32.626063],

[120.348275, 31.591534],

[121.348275, 32.591534]

]

delist = [ #站点的地理位置

[120.261926, 30.176126],

[119.980236, 31.787966],

[120.215274, 30.291919],

[119.980047, 31.788027]

]

data = numpy.array(dlist)

destion = numpy.array(delist)

# with open("local.txt", 'r') as f:

# index = -1

# for line in f.readlines():

# if len(line.strip().split(",")) < 2:

# print(len(line.strip().split(",")))

# print(line.strip().split(","))

# continue

# print(len(line.strip().split(",")))

# print(line.strip().split(","))

# if index < 43:

# data[index][0] = float(line.strip().split(",")[0])

# data[index][1] = float(line.strip().split(",")[1])

# else:

# destion[index][0] = float(line.strip().split(",")[0])

# destion[index][1] = float(line.strip().split(",")[1])

# print(data)

# print(destion)

def fit(position): #适应度函数 优化的目标（使适应度最小）

result = 0

for i, j in enumerate(position):

# print(pow(data[i]-destion[j], 2))

result += math.sqrt(sum(pow(data[i]-destion[j], 2)))

return result

# test = [1 for i in range(43)]

# print(fit(test))

# position = random.randint(1, 4)

# print(position)

# print(numpy.zeros(43))

# position = numpy.array([random.randint(0, 3) for i in range(43)])

# change = numpy.array([0.8 \* 0.1 for i in range(43)])

# speed = change + 0.3 \* random.random() \* (

# position - position) + 0.5 \* random.random() \* (

# position - position)

# print(speed)

# print(position)

# position1 = position + speed

# print(position1)

# position1 = int(position1)

# print(position1)

slove = PSO(fit, 100, 0.8, 0.3, 0.5, [0, 3]) #新建对象

best = slove.solve(1000) #当循环次数达1000时停止

print(best)

print(fit(best))

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Means算法（Python实现 有数据展示功能 但是需要加载特定个数的TXT数据文件）

#coding=utf-8

from numpy import \*

def loadDataSet(fileName):

dataMat = []

fr = open(fileName)

for line in fr.readlines():

curLine = line.strip().split(',')

fltLine = map(float, curLine)

temp = []

for i in fltLine:

temp.append(i)

dataMat.append(temp)

return dataMat

#计算两个向量的距离，用的是欧几里得距离

def distEclud(vecA, vecB):

return sqrt(sum(power(vecA - vecB, 2)))

#随机生成初始的质心

def randCent(dataSet, k):

n = shape(dataSet)[1]

centroids = mat(zeros((k,n)))

for j in range(n):

minJ = min(dataSet[:,j])

rangeJ = float(max(array(dataSet)[:,j]) - minJ)

centroids[:,j] = minJ + rangeJ \* random.rand(k,1)

return centroids

def kMeans(dataSet, k, distMeas=distEclud, createCent=randCent):

m = shape(dataSet)[0]

clusterAssment = mat(zeros((m,2)))#create mat to assign data points

#to a centroid, also holds SE of each point

centroids = createCent(dataSet, k)

clusterChanged = True

while clusterChanged:

clusterChanged = False

for i in range(m):#for each data point assign it to the closest centroid

minDist = inf

minIndex = -1

for j in range(k):

distJI = distMeas(centroids[j,:],dataSet[i,:])

if distJI < minDist:

minDist = distJI

minIndex = j

if clusterAssment[i,0] != minIndex:

clusterChanged = True

clusterAssment[i,:] = minIndex,minDist\*\*2

print(centroids)

for cent in range(k):#recalculate centroids

ptsInClust = dataSet[nonzero(clusterAssment[:,0].A==cent)[0]]#get all the point in this cluster

centroids[cent,:] = mean(ptsInClust, axis=0) #assign centroid to mean

return centroids, clusterAssment

def show(dataSet, k, centroids, clusterAssment):

from matplotlib import pyplot as plt

numSamples, dim = dataSet.shape

mark = ['or', 'ob', 'og', 'ok', '^y', '+y', 'sr', 'dr', '<r', 'pr']

for i in range(numSamples):

markIndex = int(clusterAssment[i, 0])

plt.plot(dataSet[i, 0], dataSet[i, 1], mark[markIndex])

mark = ['Dr', 'Db', 'Dg', 'Dk', '^y', '+y', 'sb', 'db', '<b', 'pb']

for i in range(k):

plt.plot(centroids[i, 0], centroids[i, 1], mark[i], markersize = 12)

plt.show()

def main():

dataMat = mat(loadDataSet('local.txt')) #码头聚类数据加载

myCentroids, clustAssing= kMeans(dataMat,5)

print(myCentroids)

means = 0

for i in clustAssing[:,1]:

means += i

print(means/(shape(dataMat)[0]))

show(dataMat, 5, myCentroids, clustAssing)

lt = [1,7,11,12,16,19,25,29,37,40] #分配给新建站点的数据加载

data = dataMat[lt]

# for i in lt:

# temp = dataMat[i].getA()

# data.append([temp[0],temp[1]])

print(data)

# data = mat(data)

cent,clust = kMeans(data,1)

show(data,1,cent,clust)

if \_\_name\_\_ == '\_\_main\_\_':

main()