所属类别	**2021 年“华数杯”全国大学生数学建模竞赛**	参赛编号

基于规划模型与智能算法的货物装运策略研究

摘要

随着航空货运业发展，如何充分利用飞机货舱装载空间和实现**经济利益最大化**成为进出口公司关注的焦点，因此，研究**货运飞机配载优化**问题具有一定的实际应用价值。

针对问题一，本文以货运飞机货舱装载货物后留下的空隙尽可能小为目标，在给定不同种类飞机数量和货源数量的前提下，求解最优的货物装运策略。首先，进行数据预处理。用 python 求解矩阵各元素的平均值作为给定的货源数量，分别求出不同机型不同货舱的容积以及 10 种货物的体积和平均密度，通过对比分析得出在题设条件下可以暂不考虑体积影响只考虑重量约束。然后，以达到最大载重量时所有货物总体积的最大值作为**目标函数**，将每个货舱受到最大载重量的限制并且每种飞机三个货舱中实际载重必须与其最大载重成比例等要求作为**约束条件**，建立**混合整数线性规划模型**并用 Lingo 求解得出，在采用最优策略时，装运货物的总体积为 626.5572 m^3，货物与货舱之间的空隙为 8831.1768 m^3。最后，本文修改了题设条件使其更**具有一般性**，并以此为前提对模型进行**改进和推广**，用**改进的贪心算法**通过 MATLAB 求解得到最优装运策略。

针对问题二，由于不同类型的集装箱有数量的限制，并且集装箱自身具有重量和体积，考虑到多目标规划的方法和可调节高度的柔性航空集装箱会给模型求解带来难度，本文在问题一整数规划模型的基础上调整参数设置，增加或修改决策变量和约束条件。先根据**定序原则、定位原则、空间分割与合并原理**对问题进行化简，借助**启发式算法**与**混合遗传算法**，分析小体积货物的装箱问题，再建立**单目标**的**整数规划模型**求解集装箱与大体积货物的机舱装载问题，将以集装箱和箱内货物之间的空隙之和尽量小以及使货舱和集装箱、箱外货物之间的空隙之和尽量小为目标的多目标规划转换为单目标规划，求解最合适的机型以及架次和各架次具体装运方案。

针对问题三，本文具体的研究了货物**运输成本**与其销售后所得收入之间的关系，从货物的种类与数量上改进装运方案，先通过对数值的判断比对确定各不同**供给-需求**条件下获得最佳利润的运送量和销售方法，再利用**正态分布**拟合等方法模拟需求量，得到可能的各需求量下的最优运载货物信息，最终利用各需求量所占总体的概率，加权计算期望，求得最佳利润所对应的货物信息。

针对问题四，根据问题三得出的销售数量服从正态分布的结论，求出各类货物不同销售数量的**概率矩阵**，并将分布函数与**可靠性**结合，以满足可靠性条件下的运送数量为决策变量，根据销售额、运输成本、生产成本以及相应销量下的概率确定期望收益的目标函数，建立整数规划模型并求解。最终得出 HW1 到 HW6 的运输量为平均销售数量，HW7 到 HW10 的运输量为平均销售数量的 70%左右，最终的总期望利润为 755.27 万元，与实际情况较为符合。

针对问题五，本文在问题四的基础上，将可靠性由 95%改为 70%，进一步探究最大销售量、最大利润、装运策略的变化，分析可靠性与最大利润的关系。

最后，本文对模型和算法进行评价与推广，得出具有高效的**全局搜索能力**，可根据实际情况自定义交叉变异概率等优点以及运行速度较慢、缺乏定性因素的考虑等缺点。

关键词：整数规划模型；混合遗传算法；贪心算法；三维装箱问题；空间分割合并；配载优化

一、问题重述

1.1 问题背景

随着全球化程度的提高，航空货运业已成为世界贸易主要推动力。如何节约货运飞机装载空间和实现货物销售利润最大化成为进出口公司亟待解决的问题，从现实约束的角度研究货运飞机配载优化这一问题具有一定的实际应用价值。

1.2 基本信息

货运飞机有大、中、小三种类型，每一种飞机均有前、中、后三个货舱，每种货物可以在一个或多个货舱中任意分布，多种货物可以混装。有 HW1-HW10 等 10 种销售货物，每件货物均为长方体。下一个周期货物的销售量是随机的，销售价格是已知的。

✱每个货舱受到最大容积、最大载重量的限制。

✱每种飞机三个货舱中实际载重必须与其最大载重成比例。

✱所有货物在装运飞机之前的所有成本为货物销售价格的 40%，如果一个周期内，产品如果不能销售出去，则按照销售价的 30%清仓甩卖。

➽*附件一*分别给出了大、中、小三种飞机前、中、后三个货舱的长、宽、高以及最大载重量。

➽*附件二*给出了不同货物的尺寸、体积（立方米）、重量（吨）、运输单价和前 50 个周期的每种货物的销售量以及下一个周期的销售价格。

➽*附件三*给出了不同型号集装箱的外部尺寸、数量和自重。

1.3 问题提出

1.3.1 问题 1 的提出：

按照前 50 个周期各种货物销售量的平均值来组织货源。假设只有大、中、小三种类型的货运飞机各一架，所有货物可以直接装入飞机。如何装运可以使得货运飞机尽量不留空隙？

1.3.2 问题 2 的提出：

按照前 50 个周期各种货物销售量的平均值来组织货源。体积在 $2m^3$ 以下的货物必须用厚度为 5 厘米的封闭的长方体塑料集装箱装载（数据见附件 3），且集装箱和货运飞机都尽量不留空隙。请分析：使用哪种机型最为合适？至少需要多少架次？各架次的具体装运方案是什么？

1.3.3 问题 3 的提出：

以集装箱尽量不留空隙、货运飞机尽量不留空隙作为货物装运的思路会影响进口公司的经济效益。为满足经济期望，应该如何调整装运方案？最佳利润是多少？

1.3.4 问题 4 的提出：

市场需求的不确定性导致一个周期内组织货源数量的风险性。安全起见，在可靠性为 95%的前提下，可以实现的最大利润值是多少？对应的货运装运策略是什么？

1.3.5 问题 5 的提出：

基于问题四，将可靠性 95%改为 70%，最大利润值和装运策略将如何改变？

二、问题分析

2.1 问题 1 的分析

问题一要求以货运飞机货舱装载货物后留下的空隙尽可能小为目标，在给定不同种类飞机数量和货源数量的前提下，求解最优的货物装运策略。首先，用 python 求解得出附件二中各种货物前 50 个周期销量的矩阵各元素的平均值作为给定的货源数量。其次，分别求出不同机型不同货舱的容积以及 10 种货物的体积和平均密度，通过对比分析得出在题设条件下可以暂不考虑体积影响只考虑重量约束。最后，将达到最大载重量时所有货物总体积的最大值作为目标函数，以货物种类、飞机种类、货舱种类等作为决策变量，将每个货舱受到最大载重量的限制并且每种飞机三个货舱中实际载重必须与其最大载重成比例等要求作为约束条件，建立混合整数线性规划模型并用 Lingo 求解。此外，还可以使用贪心法以及改进的贪心法通过 MATLAB 得到最优装运策略。

2.2 问题 2 的分析

问题二要求在货源数量与问题一相同，所有货源都需要装运，不限制各类型飞机数量，全过程只能使用一种机型，并且体积在 $2m^3$ 以下的货物必须用厚度为 5 厘米的封闭的长方体塑料集装箱装载的基础上，以集装箱和货运飞机都尽量不留空隙为目标，求解最合适的机型以及架次和各架次具体装运方案。首先，本题有 2 个目标函数，分别是使集装箱和箱内货物之间的空隙之和尽量小以及使货舱和集装箱、箱外货物之间的空隙之和尽量小，是多目标规划问题。其次，由于不同类型的集装箱有数量的限制，并且集装箱自身具有重量和体积，需要在问题一整数规划模型的基础上调整参数设置，增加或修改决策变量和约束条件。最后，考虑到多目标规划的方法和可调节高度的柔性航空集装箱会给模型求解带来难度，可以转化为单目标规划，分为两步来求解。先根据定序原则、定位原则、空间分割与合并原理对问题进行化简，借助启发式算法与混合遗传算法，分析小体积货物的装箱问题，再建立单目标的整数规划模型求解集装箱与大体积货物的机舱装载问题。

2.3 问题 3 的分析

前两问中，以集装箱尽量不留空隙、货运飞机尽量不留空隙为货物装运的原则。本问侧重考虑货物装运与经济效益之间的关系。首先，将装运方案看作两个步骤：确定需运载的货物信息，装入集装箱或飞机。据附件材料，可以看到经济利润只与货物信息有关，与使用集装箱、飞机的个数无关，与装箱方式无关。因此上述原则与得到最佳经济效益并不冲突。本问货源已知且与前问相同，即购买货物数量和所需的固定成本确定，决策中将其视为沉没成本，无需考虑。本问具体的研究了货物运输成本（变动成本）与其销售后所得收入之间的关系，对装运方案主要从货物的种类与数量上进行改进，先通过对数值的判断比对确定各不同供给-需求条件下获得最佳利润的运送量和销售方法，再利用正态分布拟合等方法模拟需求量，得到可能的各需求量下的最优运载货物信息，最终利用各需求量所占总体的概率，加权计算期望，求得最佳利润所对应的货物信息。

2.4 问题 4 的分析

问题四规定市场对产品的需求是不确定的，本环节基于前文对销量分布的检验，进一步考虑其分布的特征，可能服从正态分布，或有时间序列规律，以题目要求的 95%可靠度为条件，结合分布函数，求出在此分布下的期望收益，对于利润、销量、运输方案

之间的关系，整体直接求解，复杂度较高，因此考虑将利润与销量、销量与运输方案两组关系分别建模，前者需要综合运输成本、货物成本、销售额等因素，建立优化模型，后者可结合问题 3 中的模型改进后求解。

2.5 问题 5 的分析

问题五以问题 4 为背景，可靠性要求降低为 70%，继续使用问题 4 中的模型求出结果，进一步探究最大销售量、最大利润、装运策略的变化，分析可靠性与最大利润的关系，为企业平衡风险与收益，制定最优的运输决策提供科学支持。

三、模型假设

1、题目中给出的十种货物相互之间没有影响，每种货物可以在一个或多个货舱中任意分布，多种货物可以混装。

2、每种货物在运输过程中对货舱环境没有特殊要求，不考虑温度等外界因素对货物质量和体积的影响。

3、在货舱容积允许的情况下，货物尽可能较平均地放置在货舱中，忽略货物在货舱中的摆放方式对货舱分区载荷限重的影响。

4、假设飞机的重心符号重心包线限制，即飞机的起飞重心、无油重心和着陆重心都不得超过对应的安全重心范围。

5、不考虑弯矩、扭矩作用对货运飞机装载的影响。

6、忽略在集装单元的装卸和运输过程中货运系统受到的重压、磨损、冲击、油污侵蚀以及因未及时更换而造成的部件老化等问题对货舱容积的影响。

7、三个货舱中实际载重必须与其最大载重成比例，但不要求完全严格成比例，允许某些班次存在误差。

8、货物固定牢固，不会出现货物在集装箱中滑落导致其他货物损坏的情况。

9、不严格要求单个集装箱、机舱内的货物重心位置。

10、货物可以保持其形状和尺寸，不会由于堆叠而变形，不是危险品等特殊货物。

11、不考虑途中货物的加载与卸载。

12、假设货物可以储存在货仓，不被运输。

四、符号和变量说明

表 1 符号和变量说明表

符号	说明	量纲
$k(k=1,2,\cdots,10)$	货物种类	种
$n(n=1,2,3)$	飞机种类	种
$m(m=1,2,3)$	货舱种类	种
$s(s=1,2,\cdots,7)$	集装箱种类	种
$i(i=1,2,3,4)$	体积小于 $2m^3$ 的货物的种类	种
t	飞机的班次	次
$t_n(n=1,2,3)$	第 n 种飞机的班次	次
$N_{s,m,n}$	第 s 个集装箱的使用数量	个
$N_{k,m,n}$	在第 n 种飞机 m 种货舱中第 k 种货物的数量	箱
$N_{k,m,n}^t$	在第 n 种飞机第 t 班次 m 种货舱中第 k 种货物的数量	箱
$N_{k,m}^t$	在第 t 班次 m 种货舱中第 k 种货物的数量	箱
$N_{s,m}^t$	在第 t 班次 m 种货舱中第 s 种集装箱的数量	个
$N_{s,i}$	第 s 种集装箱中第 i 种体积小于 $2m^3$ 的货物的数量	箱
$\bar{N}_s$	第 s 种集装箱的最大数量	个
$T_k(k=1,2,\cdots,10)$	第 k 种货物的数量	箱
$T_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的数量	箱
V	货物的总体积	m^3
$V_k(k=1,2,\cdots,10)$	第 k 种货物的体积	m^3
$V_n(n=1,2,3)$	第 n 种飞机的容积	m^3
$V_s(s=1,2,\cdots,7)$	第 s 个集装箱的体积	m^3
$V_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的体积	m^3
$w_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的重量	吨
$w_{m,n}(m,n=1,2,3)$	第 n 种飞机 m 种货舱的最大载重量	吨
$w_k(k=1,2,\cdots,10)$	第 k 种货物的单位重量	吨/箱
$l_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的长	m
$r_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的宽	m
$h_i(i=1,2,3,4)$	第 i 种体积小于 $2m^3$ 的货物的高	m
$L_s(s=1,2,\cdots,7)$	第 s 种集装箱的长	m
$R_s(s=1,2,\cdots,7)$	第 s 种集装箱的宽	m
$H_s(s=1,2,\cdots,7)$	第 s 种集装箱的高	m

五、模型建立与求解

5.1 问题一的分析与求解

5.1.1 模型准备和建立

（一）数据预处理

本文首先对附件一中大、中、小三种飞机前、中、后三个货舱的长、宽、高以及最大载重量等数据进行分析。求解各机型前、中、后三个货舱的体积以及载重体积比，结果如下表所示，可以得出大型飞机的后舱最大载重量与容积的比值最小，小型飞机的中舱最大载重量与容积的比值最大。

表 2 不同机型、不同货舱数据表

飞机类型	大型	中型	小型
前舱最大载重/容积	0.0049	0.0095	0.0512
中舱最大载重/容积	0.0064	0.0091	0.0568
后舱最大载重/容积	0.0047	0.0087	0.0379
前舱尺寸（长*宽*高,m3）	2038.14	838.44	117.30
中舱尺寸（长*宽*高,m3）	2501.20	1321.78	140.76
后舱尺寸（长*宽*高,m3）	1703.52	691.03	105.57
前舱最大载重 t	10	8	6
中舱最大载重 t	16	12	8
后舱最大载重 t	8	6	4

接着，本文将附件二中不同货物的尺寸、体积（立方米）、重量（吨）、运输单价和前 50 个周期的每种货物的销售量以及下一个周期的销售价格等数据重新整理，用 python 分别计算 10 个 5×10 的矩阵个元素的平均值并四舍五入作为货源数量，并根据货物的长、宽、高分别求出 HW1-HW10 的体积和密度，最后将单个货物的货源数量分别乘以单个重量得到每种货物的总质量并将不同货物的总质量求和，最终结果如下表。

表 3 货物 HW1—HW10 相关数据表

货物名称	销货量平均值	销货量平均值(四舍五入)	体积 m3	重量 (t)	密度	总质量 (t)
HW1	119.32	119	7.59	2.1	0.28	249.90
HW2	367.54	368	1.16	0.2	0.17	73.60
HW3	361.10	361	5.71	0.7	0.12	252.70
HW4	364.32	364	5.07	1.8	0.36	655.20
HW5	247.38	247	2.41	1.3	0.54	321.10
HW6	307.34	307	0.71	0.3	0.42	92.10
HW7	611.18	611	0.21	0.23	1.07	140.53
HW8	2993.14	2993	2.47	1.2	0.49	3591.60
HW9	617.40	617	2.87	0.9	0.31	555.30
HW10	1225.26	1225	1.50	0.3	0.20	367.50
						6299.53

从表中可以看出，单个重量最小的货物为 HW2，单个重量最大的货物为 HW1；密度最

小的货物为 HW3，密度最大的货物为 HW7；所有货物的总质量之和为 6299.53t。通过观察附件二中不同货物的长、宽、高，可以得出只有个别种类的货物如 HW2 的形状为正方体，其余均为长方体。

在问题一中，由于货物的体积和货舱的容积以及货物的质量和货舱的最大载重量已知，可以通过比较分析两两之间的相互数量关系来粗略估计建模的方向。根据密度的定义可以得出在载重量一定的条件下，密度越小，所占空间的体积越大，反之亦然。飞机货舱的最大载重与容积的比值越大，说明在载重量相同的情况下，此货舱需要更小的容积，反之亦然。首先，假设将密度最小的货物 HW3 放入最大载重与容积的比值最大的小型飞机中舱；其次，分别计算达到最大载重量与最大容积时所需货物 HW3 的体积和质量，最后得出达到最大载重只需 11 个货物 HW3 和 1 个货物 HW6，根据*表 2* 和*表 3*，此时货物总体积为 $5.71\times 11+0.71\times 1=63.52m^3$，远小于 $140.76m^3$，而达到最大容积至少需要 24 个货物 HW3。因此，在问题一的假设条件下，无论何种货物、何种货舱，重量对装配空间的影响远大于体积，可以暂时将体积忽略不计。

（二）混合整数线性规划模型

假设共有 k 种货物，则 $k=1,2,\cdots,10$。假设用 n 来代表飞机种类，用 m 来代表货舱种类，则可得到公式（1）和（2）：

$$n=\begin{cases}1, & \text{大型飞机}\\ 2, & \text{中型飞机}\\ 3, & \text{小型飞机}\end{cases} \tag{1}$$

$$m=\begin{cases}1, & \text{飞机前舱}\\ 2, & \text{飞机中舱}\\ 3, & \text{飞机后舱}\end{cases} \tag{2}$$

接着，本文用 $N_{k,m,n}$ 表示第 k 种货物在第 n 种飞机 m 种货舱的数量，用 $T_k(k=1,2,\cdots,10)$ 表示第 k 种货物的数量，用 $V_k(k=1,2,\cdots,10)$ 表示第 k 种货物的体积，用 $w_{m,n}(m,n=1,2,3)$ 表示第种 n 飞机 m 种货舱的最大载重量。

（1）约束条件

由于每个货舱受到最大容积、最大载重量的限制，可以得到约束条件①：

$$\sum_{k=1}^{10}(N_{k,m,n}\times w_k)\le w_{m,n}(m,n=1,2,3) \tag{3}$$

由于每种飞机三个货舱中实际载重必须与其最大载重成比例，可得约束条件②：

$$\frac{\sum_{k=1}^{10}(N_{k,1,n}\times w_k)}{w_{1,n}}=\frac{\sum_{k=1}^{10}(N_{k,2,n}\times w_k)}{w_{2,n}}=\frac{\sum_{k=1}^{10}(N_{k,3,n}\times w_k)}{w_{3,n}}(n=1,2,3) \tag{4}$$

由于每种货物的累计数量小于每种货物的总数，可得约束条件③：

$$\sum_{m=1}^{3}\sum_{n=1}^{3}N_{k,m,n}\le T_k(k=1,2,\cdots,10) \tag{5}$$

因为第 k 种货物在第 n 种飞机 m 种货舱的数量必为正整数，可得约束条件④：

$$N_{k,m,n} \in Z \tag{6}$$

（2）目标函数

通过数据预处理中的对比分析得出在题设条件下可以暂不考虑体积影响只考虑重量约束，为了使货物与货舱之间的空隙尽可能小，将达到最大载重量时所有货物总体积的最大值作为目标函数：

$$\max z = \sum_{k=1}^{10}\sum_{m=1}^{3}\sum_{n=1}^{3}(N_{k,m,n} \times V_k) \tag{7}$$

（3）建立模型

$$\max z = \sum_{k=1}^{10}\sum_{m=1}^{3}\sum_{n=1}^{3}(N_{k,m,n} \times V_k)$$

$$s.t.\begin{cases} \sum_{k=1}^{10}(N_{k,m,n} \times w_k) \le w_{m,n}(m,n=1,2,3) \\ \dfrac{\sum_{k=1}^{10}(N_{k,1,n} \times w_k)}{w_{1,n}} = \dfrac{\sum_{k=1}^{10}(N_{k,2,n} \times w_k)}{w_{2,n}} = \dfrac{\sum_{k=1}^{10}(N_{k,3,n} \times w_k)}{w_{3,n}}(n=1,2,3) \\ \sum_{m=1}^{3}\sum_{n=1}^{3} N_{k,m,n} \le T_k(k=1,2,\cdots,10) \\ N_{k,m,n} \ge 0\text{且为整数} \end{cases} \tag{8}$$

5.1.2 模型的求解

利用 lingo 软件求解得到结果整理如下（边框加粗的部分为结果）：

表 4　问题一的结果

求和项:数量	货物种类										总计	质量
机型与舱型	HW1	HW2	HW3	HW4	HW5	HW6	HW7	HW8	HW9	HW10		
前舱	0	3	33	0	0	0	0	0	0	1	37	24
大型飞机	0	1	14	0	0	0	0	0	0	0	15	10
中型飞机	0	0	11	0	0	0	0	0	0	1	12	8
小型飞机	0	2	8	0	0	0	0	0	0	0	10	6
中舱	0	7	49	0	0	0	0	0	0	1	57	36
大型飞机	0	3	22	0	0	0	0	0	0	0	25	16
中型飞机	0	4	16	0	0	0	0	0	0	0	20	12
小型飞机	0	0	11	0	0	0	0	0	0	1	12	8
后舱	0	3	24	0	0	0	0	0	0	2	29	18
大型飞机	0	0	11	0	0	0	0	0	0	1	12	8
中型飞机	0	2	8	0	0	0	0	0	0	0	10	6
小型飞机	0	1	5	0	0	0	0	0	0	1	7	4
总计	0	13	106	0	0	0	0	0	0	4	123	78

最优化后得出当三种类型的飞机只有一架并且只允许一个班次时，使空隙最小的的装运方法是只装货物 HW2、HW3、HW10，具体分配细节见上表。结合上文的表二，可以看出货物 HW2、HW3、HW10 是十种货物中密度最小的，恰好验证了前文的假设和推论。

5.1.3 模型的改进与推广

由于题目一中要求只装 3 个货机，不考虑将货物装完，即只装入部分货物。而在现实生活中往往会遇到给定运货总量求解空隙最小的装运方案的情况，此时需要计算至少需要多少班次以及各个班次的装运方式。因此，可以在前文所建模型的基础上进一步改进和推广，假设其它条件不变，大、中、小机型的飞机只能有一辆，可以飞多个班次，货源数量与问题一相同并且要求全部装运，2 种改进和推广的算法和模型如下。

（一）改进的贪心算法

为了更简单、迅速地得到最优解，本文采用贪心算法，在求解最小空隙时，每一次都选择使当前班次货物装运后留下空隙最小的货物装运方法，不从整体最优上加以考虑，即每一次都得到某种意义上的局部最优解。货运飞机货舱装载货物后留下的空隙尽可能小的整体最优解可通过一系列局部的最优解的选择达到，并且每次的选择可以依赖以前作出的选择，但不依赖于后面要作出的选择。

此算法的特点是一步一步地进行，常以当前情况为基础根据某个优化测度作最优选择，而不考虑各种可能的整体情况，省去了为找最优解要穷尽所有可能而必须耗费的大量时间。本文采用自顶向下的方法，即从大型飞机到中型飞机再到小型飞机，从前舱到中舱再到后舱，从第一个班次开始，逐步往后，以迭代的方法做出相继的贪心选择。所谓“贪心的选择”，在本文中主要指先装密度小的货物。所有机型的所有货舱最大载重量给定，货物的密度越小，载重达到上限时货物所占空间就越大，因此从局部来看，每次都先装运当前密度最小的货物是最好的选择。每做一次贪心选择，就将所求问题简化为一个规模更小的子问题，通过每一步贪心选择，可得到问题一的一个最优解。

但是由于本题中的货物不可以无限细分，使用常规的贪心算法得到的结果存在一定误差，还未达到最优，仍然存在进一步优化的空间。如下表所示，以密度最小的货物 HW3 为例，按照贪心算法的原则，最开始在大型飞机的前舱不断加入 HW3（单个重量 0.7t）直至达到载重上限 10t。然而，在装完 14 个 HW3 时，实际载重已达 9.8t。此时不可能再增加 HW3，但是还有 0.2t 的载重剩余空间，因此，本文使用改进的贪心算法，用 HW2（单个重量 0.2t）补充，使最终的实际重量刚好等于货舱最大载重量。

表 5　改进贪心法数据例表（部分）

货物数量	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
增加数量	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.2
累积重量	0.7	1.4	2.1	2.8	3.5	4.2	4.9	5.6	6.3	7	7.7	8.4	9.1	9.8	10

下图为改进的贪心算法的流程图。基于货物信息 F 和飞机信息 M，本文首先将各类货物的重量按照需求量展开，将不同种类的货物按照平均密度从低到高进行排序，密度低的先装入货舱，如果无法完全达到最大载重量还有微量的剩余载重空间，则优先填充

质量较小的货品直到实际载重等于或无限接近于货舱的最大载重量，最后输出飞机运载结果的矩阵。

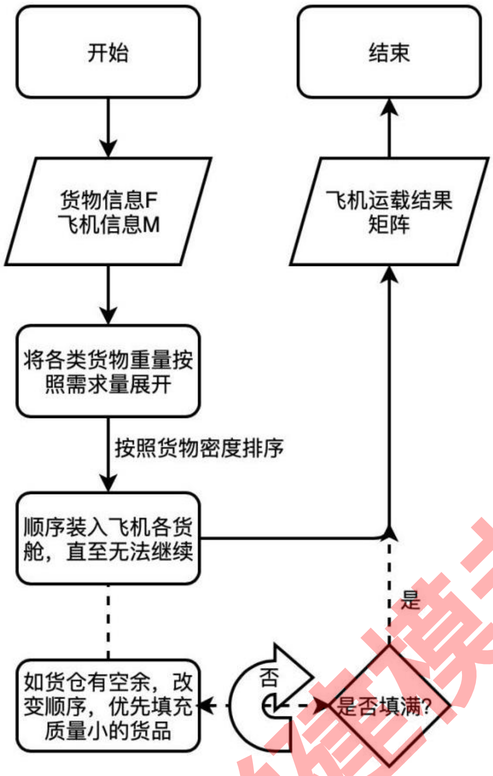

图 1　改进的贪心算法流程图

根据改进的贪心算法，计算得出至少需要 83 次才能将所有货物运完。

（二）推广的混合整数线性规划模型

推广到大、中、小机型的飞机只能有一辆，可以飞多个班次的情况：货源数量与问题一相同，货物的体积不变，货舱的容积随着班次的增加而增加，要使货舱与货舱内货物的空隙之和尽可能小，需尽可能减小运输的班次。因此，优化的目标是使所有班次所有飞机全部货舱容积的总和与总货物的体积之差最小。

设 V 为货物的总体积，$t_n(n=1,2,3)$ 为第 n 种飞机的班次，$V_n(n=1,2,3)$ 为第 n 种飞机的容积，修改后的目标函数如下：

$$\min z=\sum_{n=1}^{3}(t_n\times V_n)-V \tag{9}$$

$$V=\sum_{k=1}^{10}(V_k\times T_k) \tag{10}$$

将上文的公式（5）稍作修改，得到新的约束条件：

$$\sum_{m=1}^{3}\sum_{n=1}^{3}N_{k,m,n}=T_k\,(k=1,2,\cdots,10) \tag{11}$$

整理后，建立优化模型如下：

$$\min z=\sum_{n=1}^{3}(t_n\times V_n)-\sum_{k=1}^{10}(V_k\times T_k)$$

$$s.t.\begin{cases}\sum_{k=1}^{10}(N_{k,m,n}\times w_k)\le w_{m,n}(m,n=1,2,3)\\ \dfrac{\sum_{k=1}^{10}(N_{k,1,n}\times w_k)}{w_{1,n}}=\dfrac{\sum_{k=1}^{10}(N_{k,2,n}\times w_k)}{w_{2,n}}=\dfrac{\sum_{k=1}^{10}(N_{k,3,n}\times w_k)}{w_{3,n}}(n=1,2,3)\\ \sum_{m=1}^{3}\sum_{n=1}^{3}N_{k,m,n}=T_k(k=1,2,\cdots,10)\\ N_{k,m,n}\ge 0\text{且为整数}\end{cases}\tag{12}$$

根据推广的混合整数线性规划模型，可以得出至少需要 81 次才能将所有货物运完，且 81 个班次无先后顺序要求，可以任意交换顺序。将装运方式进行整理得到下表：

表 6　不同货物组合装运数据表

组合序号	组合方式	组合数量	数量求和	组合重量
1	HW3+HW6	307	307	1
2	HW2+HW4	364		
3	HW3+HW5	54	422	2
4	HW2+2HW9	4		
5	HW1+HW9	119		
6	2HW8+2HW10	612	976	3
7	HW8+2HW9	245		
8	HW5+2HW8+HW10	1	1	4
9	2HW5+2HW8	96	96	5
10	5HW8	266	266	6
11	HW7	611	611	0.23

根据上表，在 81 个班次的运输过程中：有 78 次是以 1-10 的组合方式装配运输的，每个货舱均达到达到载重上限，且每个班次的每架飞机实际载重与其各个货舱的最大载重完全严格成比例；有 3 次情况较为特殊。

表 7　第 79 班次实际载重数据

	大型飞机	中型飞机	小型飞机
前舱实际载重	9.20	7.36	5.52
中舱实际载重	14.72	11.04	7.36
后舱实际载重	7.36	5.52	3.68

表 8　第 80 班次实际载重数据

	大型飞机	中型飞机	小型飞机
前舱实际载重	8.05	7.36	4.83
中舱实际载重	12.88	11.04	6.44
后舱实际载重	6.44	5.52	3.22

如表 7 和表 8 所示，这 2 种装载方式特殊的原因不但满足每架飞机实际载重与其各个货舱的最大载重完全严格成比例，并且这 2 个班次所有的货物都是组合之后剩下的单个重量为 0.23t 的货物 HW7。

表 9　第 81 班次实际载重数据分解

	大型飞机	中型飞机	小型飞机	大型飞机	中型飞机	小型飞机
前舱实际载重	10*1t	8*1t	2*3t	10	8	6
中舱实际载重	5*3t+1t	4*3t	8*1t	16	12	8
后舱实际载重	8*1t	2*3t	1*1t+13*0.23t	8	6	3.99

如上表所示，这种装载方式特殊的原因是它由 1-11 的组合方式装配货舱，并不是完全严格成比例，有 0.01t 的误差，载重比是$10:8:6:16:12:8:8:6:3.99$。

5.2　问题二的分析与求解

问题二要求货源为 50 次销售的平均值，$2m^3$ 以下的货物全部装入集装箱，将所有货物通过一种飞机多班次运输。根据表 3，单位体积小于$2m^3$的货物只有 4 个，分别为 HW2、HW6、HW7、HW10。若直接求解，约束条件过多，时间复杂度必然较高，体积小于$2m^3$货与其他货物，并无较强联系，因此本文将问题分解为：

$$\begin{cases}\text{小体积货物的装箱问题}\\ \text{集装箱与大体积货物的机舱装载问题}\end{cases}$$

5.2.1 基于混合遗传算法的三维装箱模型

（一）模型建立

（1）目标函数：

装箱问题目标可以选用一个或者多个，不同目标之间也存在着相互的联系，通常有体积利用率最高、载重利用率最高、运输成本最低、重量平衡等目标。本文中的目标函数为集装箱体积利用率最优：

$$\max z=\frac{\sum_{i=1}^{4}(V_i\times T_i)}{\sum_{s=1}^{7}(V_s\times N_s)} \tag{13}$$

其中分子为小于$2m^3$货物的总体积，分母为使用的集装箱货物总体积。第s种集装箱的体积表示为$V_s(s=1,2,\cdots,7)$，第s种集装箱的使用数量表示为N_s，第i种体积小于$2m^3$的货物的体积表示为$V_i(i=1,2,3,4)$，第i种体积小于$2m^3$的货物的数量表示为$T_i(i=1,2,3,4)$。

（2）约束条件：

在此问题中，货物尺寸、重量、形状、数量以及集装箱的规格等可能是不同的，这在一定程度上增加了集装箱装载的复杂性。针对以上对集装箱装载问题分类以及复杂性的讨论，集装箱装载的约束做如下：

①货物总体积约束：

集装箱装载货物的过程中，空间不断缩小，集装箱体积与货物体积是装载可行性的主要标准。

$$\sum_{i=1}^{4}(N_{s,i}\times V_i)<V_s(s=1,2,\cdots,7) \tag{14}$$

其中 $N_{s,i}$ 表示第 s 种集装箱中第 i 种体积小于 $2m^3$ 的货物的数量。

②货物装载三维尺寸约束：

$$\begin{cases}0\le x_{i,s}+l_i\le L_s\\ 0\le y_{i,s}+r_i\le R_s \quad (s=1,2,\cdots,7)\\ 0\le z_{i,s}+h_i\le H_s\end{cases} \tag{15}$$

其中，$l_i(i=1,2,3,4)$ 为第 i 种体积小于 $2m^3$ 的货物的长；$r_i(i=1,2,3,4)$ 为第 i 种体积小于 $2m^3$ 的货物的宽；$h_i(i=1,2,3,4)$ 为第 i 种体积小于 $2m^3$ 的货物的高；$L_s(s=1,2,\cdots,7)$ 为第 s 种集装箱的长；$R_s(s=1,2,\cdots,7)$ 为第 s 种集装箱的宽；$H_s(s=1,2,\cdots,7)$ 为第 s 种集装箱的高；$x_{i,s}$、$y_{i,s}$、$z_{i,s}$ 为在第 s 种集装箱中第 i 种体积小于 $2m^3$ 的货物摆放位置参考坐标。

③货物总数量约束：

$$\sum_{s=1}^{7}N_{s,i}=T_i(i=1,2,3,4) \tag{16}$$

（3）模型结果：

$$\max z=\frac{\sum_{i=1}^{4}(V_i\times T_i)}{\sum_{s=1}^{7}(V_s\times N_s)}$$

$$s.t.\begin{cases}\sum_{i=1}^{4}(N_{s,i}\times V_i)<V_s\\ 0\le x_{i,s}+l_i\le L_s\\ 0\le y_{i,s}+r_i\le R_s \quad (s=1,2,\cdots,7)(i=1,2,3,4)\\ 0\le z_{i,s}+h_i\le H_s\\ \sum_{s=1}^{7}N_{s,i}=T_i\end{cases} \tag{17}$$

（二）模型求解

三维装箱问题本身具有一定的复杂性，同时为了满足实际应用中的需求，必须使用启发式算法来生成有效的装载方案。优秀的启发式算法应具备在较短的时间内生成高质量的解，同时还应具有适应不同需求的灵活性。本文基于以下原则为最优解的生成过程，一定程度上能够提高收敛速度，提高适应度。

（1）定序规则

在此规则包括体积递减、最长边递减。以货物最长边递减规则为例，首先比较货物最长边，如果最长边相同则比较次长边，然后沿着集装箱的长度方向进行装载；如果货物的最长边和次长边都相同，依照降序的规则进行装填。

（2）定位规则

在待装载货物遵循一定的先后次序逐个装箱的过程中，需要相应的定位规则确定其摆放位置。定位规则包含多种方法，首先将货物放置在集装箱的一角，然后沿着边依次装载，最后再填满中心的另一角，同时也符合实际问题中的装箱过程。因此本文采用占角策略开展装载研究。

（3）空间合并规则

在实际编写算法的过程中，会存在一些无法利用的零散空间，所以要以一定的方法进行整合。空间合并是为了充分利用零散空间而形成更大的待装载空间。这样不但可以提高集装箱的空间利用率，同时也将改善货物的装载效果。题目中 HW7 与集装箱质量的精度为 0.01t，而其他货物质量的精度均为 0.1t，考虑到若集装箱装箱后的质量精度若为 0.01t，不利于第二部分装仓过程的质量组合，同时 HW7 的体积较小，所以本文首先将集装箱与少量的 HW7 相组合，使装箱后总质量精度为 0.1t。

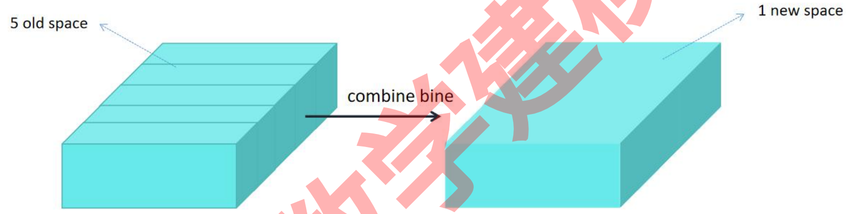

图 2　货物合并示意图

（4）启发式算法步骤：

根据三空间分割启发式算法的设计理念，结合定序规则、定位规则、空间分割规则以及空间合并规则，使货物的装载过程有序可控。算法具体步骤如下：

step 1　导入集装箱与货物基本数据，通过函数变量定义货物种类数、货物数量总数、集装箱序号，根据空间编码规则，初始化集装箱剩余空间。

step 2　对货物的摆放位置进行编码： $C=(i,j,x,y,z,l,w,h)$ 其中，i 为集装箱的箱号，j 为货物的种类号，x,y,z 为货物放置的起始坐标点，l,w,h 为货物的长、宽、高。

step 3　对待放入集装箱货物的尺寸与集装箱最大承载量进行判断，货物的长、宽、高小于集装箱剩余空间的长、宽、高，货物重量之和小于集装箱的最大承载量。

step 4　当货物满足上述装载要求时，按照上述的编码方式进行装载。装载完毕后，根据空间分割规则产生三个子空间，分别为前空间、右空间和上空间。每次生成剩余空间集合后，都要根据空间合并规则对空间进行合并操作。

step 5　对剩余空间中的 Z 坐标进行排序，进而确定空间的高低，确保货物是从下往上进行装载。

step 6　返回步骤 3，依次加载后续货物，直至货物序列集合为空。

step 7　当集装箱剩余空间不能满足货物装载，就要用下一个集装箱。

step 8　最后得出装载结果和剩余空间的数据。

根据以上步骤说明，三空间分割启发式算法流程图如下图所示。

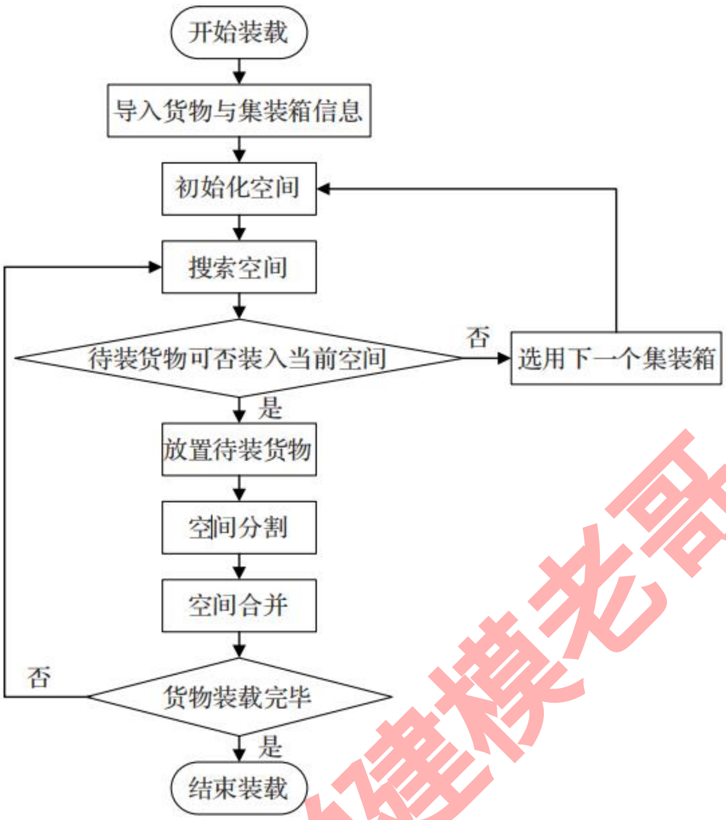

图 3　启发式算法流程图

（三）结果分析

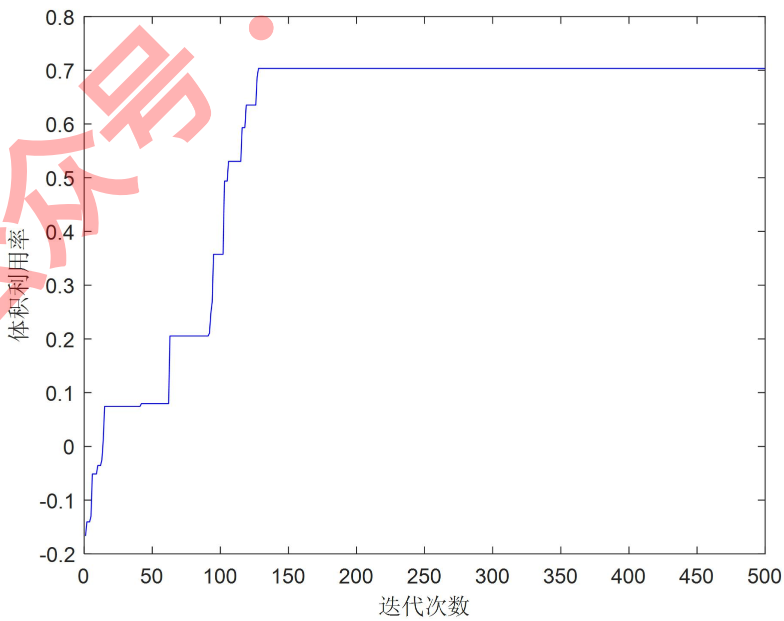

图 4　适应度曲线

对题目中所给数据进行测试，通过上图适应度曲线可以看出，本文设计的遗传算法能够在较快的时间内搜索到全局最优解，迭代 200 次左右，从而证明此算法具有很强的收敛性。 使用 Y4 集装箱，所有货物装运完成后，其最大装箱体积利用率为 71.57%，具体结果如下表所示。

表 10　集装箱使用列表

序号	装法数量	货物总数	货物重量	货物体积	体积利用率（%）	货物数量			
						HW10	HW7	HW6	HW2
装法 1	25	29	7.58	13.44	85.05	1	16	12	0
装法 2	1	31	7.76	12.68	80.21	2	22	7	0
装法 3	10	24	5.54	11.48	72.66	2	18	0	4
装法 4	1	16	3.77	11.06	69.97	3	7	0	2
装法 5	162	7	1.90	9.82	62.11	5	0	0	2
装法 6	60	6	1.80	9.00	56.95	6	0	0	0
装法 7	1	5	1.50	7.50	47.46	5	0	0	0

由于下一个集装箱是在当先集装箱满载后才进行装载，因此其装载结果不能对本文算法说明问题，如果考虑更多货物数据情况下，能够满足第二个集装箱全部装满，可能会得到一个较好的体积利用率。通过 VFD 可视化工具，本文将货物的摆放位置进行了可视化展示，下图即为本次得出的部分三维装载图示，完整结果见附录。

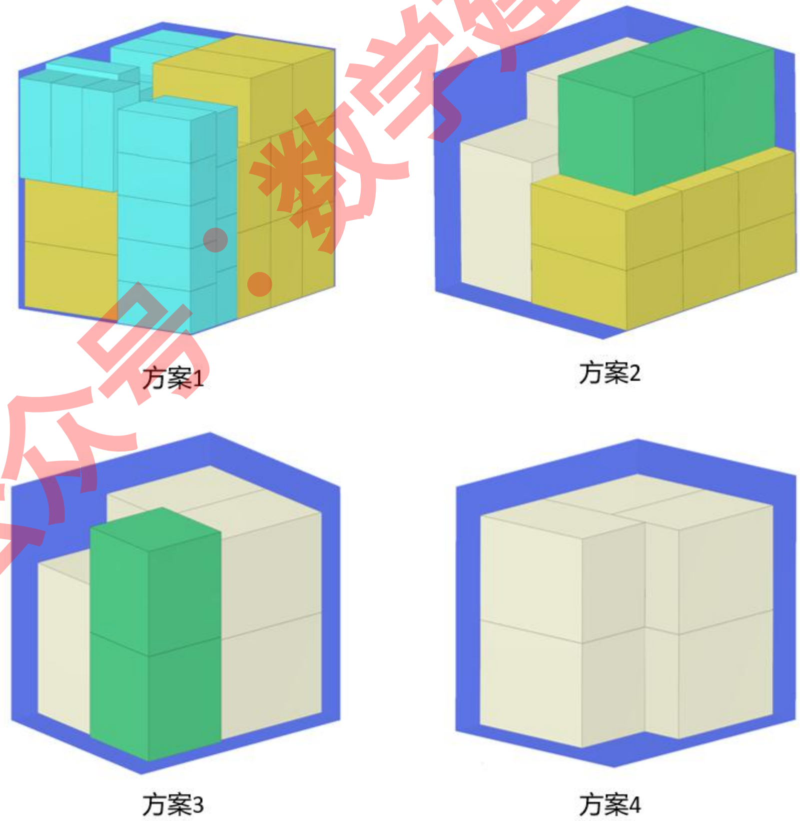

图 5　集装箱装载三维图

5.2.2 基于整数规划的货舱装运模型

上一环节中，得到 4 种货物的装箱数据，将集装箱视为组合后的新商品，与其余的六种货物共同装入飞机中，将问题转化为 7 种货物的机场装仓，基于问题 1 的求解过程，引入周期变量，建立整数规划模型，用 lingo 求解。

（一）模型建立

（1）目标函数：

$$\min t_0 \tag{18}$$

（2）约束条件：

当选择大飞机时，将上一环节中的装法 1、3、5、6 定义为第 7、8、9、10 种货物，大飞机的每个货舱都有最大载重量的限制，由此得到约束条件①：

$$\sum_{k=1}^{10}(N_{k,m}^{t}\times w_k)\le w_{1,m} \qquad (m=1,2,3) \quad (t=1,2,\cdots,t_0) \tag{19}$$

飞机的每个货舱实际载重量要与最大载重量成比例，由此得到约束条件②：

$$\frac{\sum_{k=1}^{10}(N_{k,1}^{t}\times w_k^{''})}{w_{1,1}}=\frac{\sum_{k=1}^{10}(N_{k,2}^{t}\times w_k^{''})}{w_{2,1}}=\frac{\sum_{k=1}^{10}(N_{k,3}^{t}\times w_k^{''})}{w_{3,1}} \tag{20}$$

飞机所有班次各个货舱装载的货物总量之和等于总的货物数量，得到约束条件③：

$$\sum_{t=1}^{t_0}\sum_{m=1}^{3}N_{k,m}^{t}=T_k^{'} \qquad (k=1,2,\cdots,7) \tag{21}$$

使用的所有集装箱的数量之和不能超过集装箱的总数，可得到约束条件④：

$$\sum_{k=7}^{10}\sum_{m=1}^{3}N_{k,m}^{t}\le \bar{N} \tag{22}$$

在第 t 班次 m 种货舱中第 s 种集装箱的数量是正整数，得到约束条件⑤：

$$N_{k,m}^{t}\ge 0,\text{且为整数} \tag{23}$$

（3）模型建立

$$\min t_0$$

$$s.t.\begin{cases}\sum_{k=1}^{10}(N_{k,m}^{t}\times w_k)\le w_{1,m} \qquad (m=1,2,3) \quad (t=1,2,\cdots,t_0)\\ \sum_{t=1}^{t_0}\sum_{m=1}^{3}N_{k,m}^{t}=T_k^{'} \qquad (k=1,2,\cdots,7)\\ \frac{\sum_{k=1}^{10}(N_{k,1}^{t}\times w_k^{''})}{w_{1,1}}=\frac{\sum_{k=1}^{10}(N_{k,2}^{t}\times w_k^{''})}{w_{2,1}}=\frac{\sum_{k=1}^{10}(N_{k,3}^{t}\times w_k^{''})}{w_{3,1}}\\ \sum_{k=7}^{10}\sum_{m=1}^{3}N_{k,m}^{t}\le \bar{N}\\ N_{k,m}^{t}\ge 0,\text{且为整数}\end{cases} \tag{24}$$

（二）模型求解

通过 Lingo 软件进行求解，得出各个货舱每班次运送的货物重量小于限重，满足重量比例约束，部分货物需要完全装入集装箱中，求解出总的空间利用率为 65.34%的运输方案，完成运送任务至少需要约 432 架次中型飞机。

5.3 问题三的分析与求解

5.3.1 模型准备

问题三将求解的目标由集装箱尽量不留空隙、货运飞机尽量不留空隙转移到经济利益最大化，将附件三中的数据整理后得到下表：

表 11 不同机型不同货舱最多可以装下不同尺寸集装箱的个数

航空集装箱名称	外部尺寸/m	大飞机			中飞机			小飞机		
		前舱	中舱	后舱	前舱	中舱	后舱	前舱	中舱	后舱
Y1 Y2 Y4 Y5	2.991*2.438*2.438	90	108	75	40	60	30	4	4	3
Y7	2.235*2.743(货物高度≤2.0)	144	180	126	50	72	36	5	6	4
Y3 Y6	1.456*2.438*1.9	252	320	200	100	168	80	10	12	8

由上表可得，Y1、Y2、Y4、Y5 的外部尺寸相同，Y3、Y6 的外部尺寸相同，大飞机的前舱最多可以装下 90 个外部尺寸为 2.991*2.438*2.438 的集装箱，小飞机的后舱最多可以装下 8 个外部尺寸为 1.456*2.438*1.9 的集装箱。

5.3.2 模型的建立

第三问中，有公式：利润=正常销售收入+倾销收入-变动成本(VC)-固定成本。其中，货源即各货物的种类数量确定（固定成本确定）。本目中变动成本为运输价格，只与待运输的各货物的种类和数量有关，即 $VC = f(T_k)(k=1,2,\cdots,10)$，$T_k$ 不确定故 VC 不确定。

本问对货源与待运输货物量进行了区分，认为待运输货物量是小于等于货源的任意整数，这与前二问中，货源全部被运输不同。设想，若二者相等，则有固定成本、变动成本均确定。购买者的需求量一旦被确定，该利润固定，且很可能是非最大利润。需结合各可能的销售额被选中为确定的销售额的概率，合理的选择待运输的货物数量。

根据正态性检验，假设货物需求量分布是基于相对应历史数据的正态分布函数，且需求量不确定。对各种货物而言，均可能出现以下 2 种情况：

1、需求量>=待运输货物数，货物供不应求（供求相等），所有运输货物全部被销售，取得该运输量下的最大利润（利润随运输量单调递增）。因而，此种情况下，待运输货物数应在货源允许的范围内尽可能的增多，与需求量之差尽可能的小，力求二者相等。

2、需求量<待运输货物数，货物供过于求，存在待运输货物数与需求量之差个货物需要倾销。

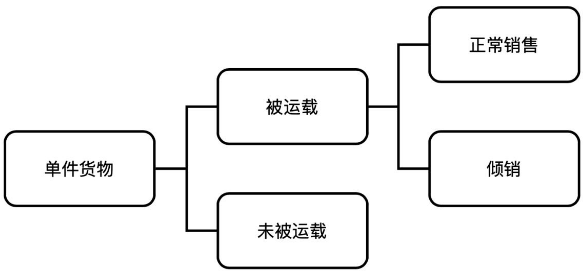

图 6 单个货物可能出现的情况

对于单件货物而言，有正常销售，倾销，未被运载三种情况。毋庸置疑，正常销售会使得这单件货物的利润最大。但由于需求量在一定范围内都有相应存在的概率。货品往往不能被正常销售。此情况下，依据利益最大原则，应在倾销和未被运载两种情况中，选择损失最小的。计算各种货物单件的（倾销收入-变动成本），暂记为 P，下表：

表 12　倾销收入-变动成本

种类	HW1	HW2	HW3	HW4	HW5	HW6	HW7	HW8	HW9	HW10
P	-400	0	150	-460	60	80	0	140	150	100

可以看出，部分种类货物只要被销售，就会有利可图（不考虑固定成本）。故认为这样种类的商品的全部货源应被运载销售。而部分种类货物的运输成本（变动成本）较大，运载后倾销得不偿失，部分此类商品可能不被运载。

5.3.3 模型的求解

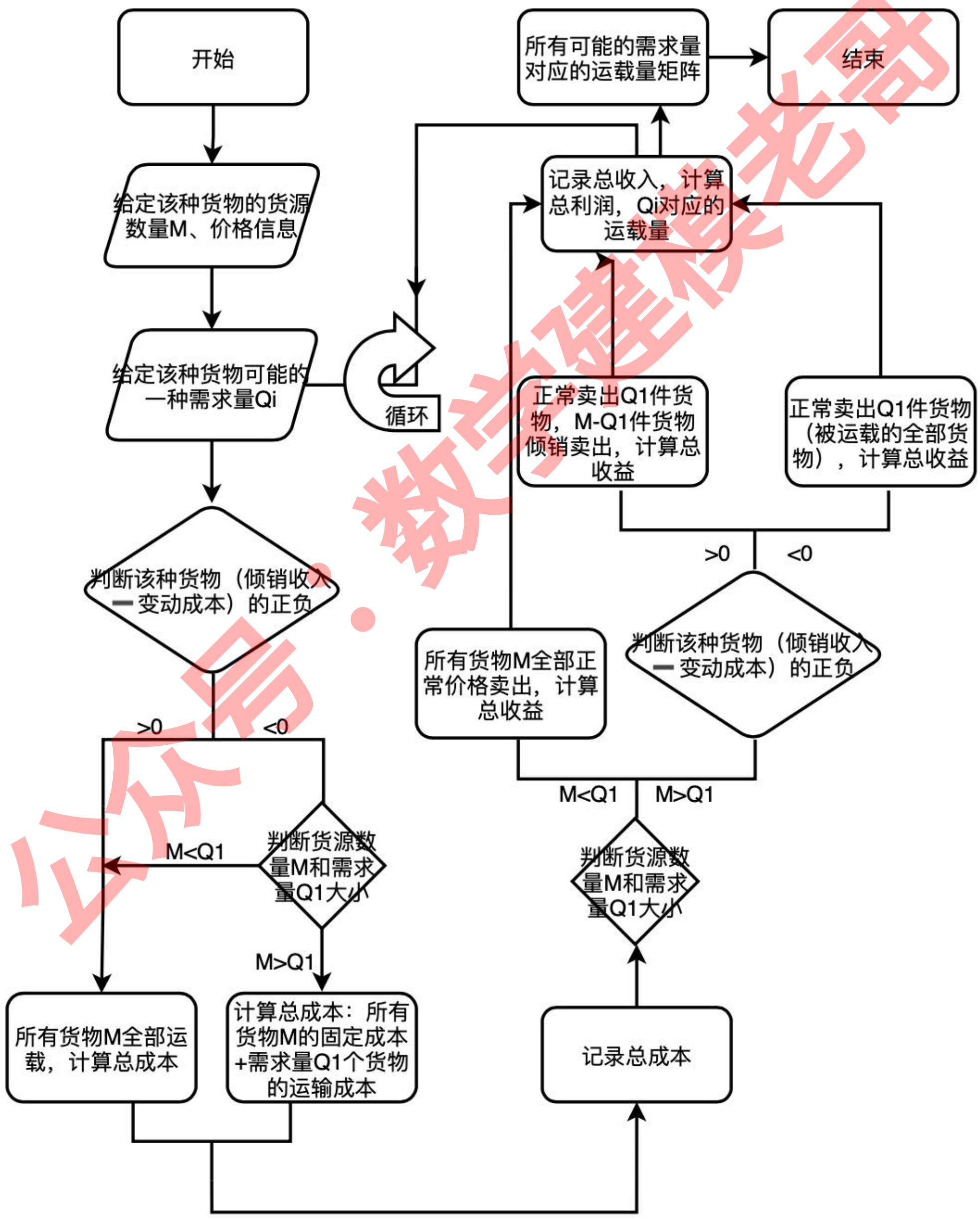

图 7　问题三的求解思路

准备工作：

1.利用单样本正态分布 Lilliefors 检验，Jarque-Bera 检验，Anderson-Darling 检验，检验每种货物历史需求量是否满足正态性假设，对比这三个结果，可以得出该数据与正态分布较为吻合。

2.对每种货物历史需求量进行置信度为 0.05 的正态分布拟合，求得均值 μ 和方差 σ 的估计值及置信区间。

3.认为[[μ -4* σ]，[μ +4* σ]]（其中内层中括号代表向外取整）包含的 n 个整数为可能取到的需求量 $X=(x_1,x_2,\ldots,x_n)'$。

依下述流程，得到各个需求量下对应的，使得利润最大的运载量构成的矩阵 $Z=(z_1,z_2,\ldots,z_n)'$。根据各需求量发生的概率不同，计算对应的运载量的期望 E(Z)，以此作为使得总利润最大的运载方式。

$$E(Z)=\sum_{i=1}^{n}P(x_i)z_i\text{，其中 }P(x)=f(x|\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{25}$$

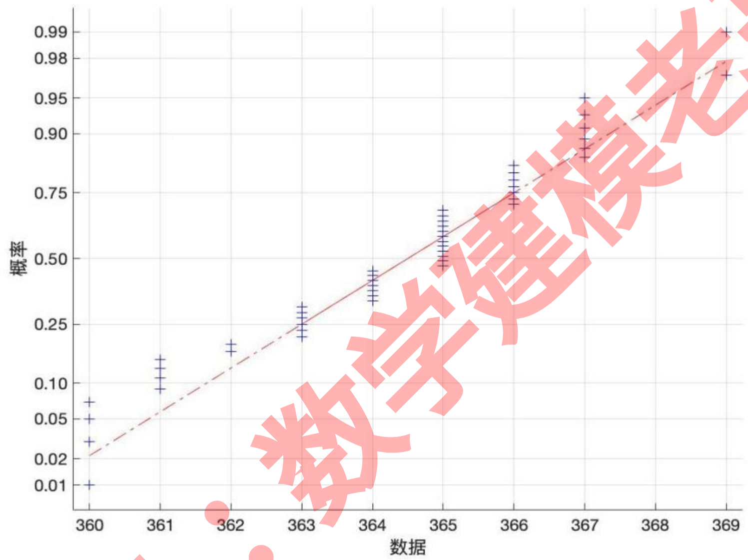

图 8 正态概率图

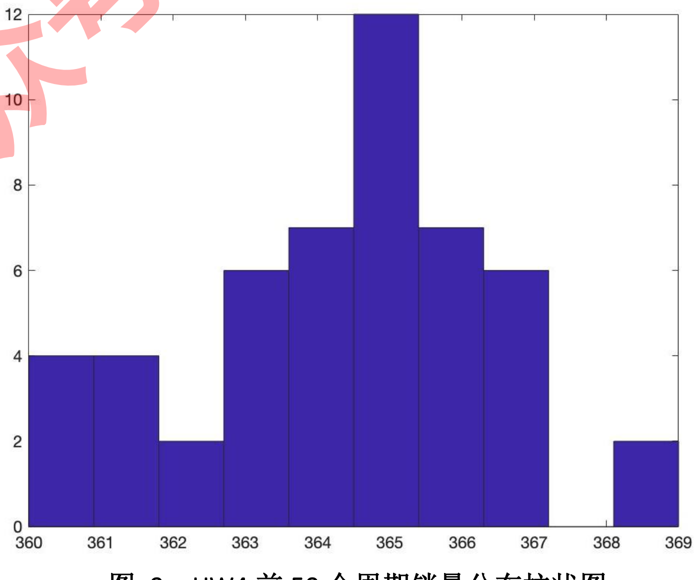

图 9 HW4 前 50 个周期销量分布柱状图

对 E(Z)附近的一些值进行枚举，进一步求解，减少误差。待运载数量代入问题 2 中模型，即可求解运送方案。

5.3.4 结果分析

结果如下：

表 13　问题三的结果

货物种类	HW1	HW2	HW3	HW4	HW5	HW6	HW7	HW8	HW9	HW10
货源数量	119	368	361	364	247	307	611	2993	617	1225
待运载数量	119	368	361	363	247	307	595	2993	617	1225
最大的利润期望	235,809	164，831	968，240	788，605	323,830	134,484	79,270	3,727,280	645,167	738,859

总的最大的利润期望为：7806825 元。将将待运载数量代入问题 2 中模型，即可求解运送方案。

5.4　问题四的分析与求解

5.4.1 模型准备

题目中的可靠性要求，定义为供货数量大于等于销售数量的概率，跟据前 50 次销售数据的统计，需求量服从正态分布，在可靠性 95%的条件下，可得到需求量向量为：以次向量为供货数量，结合销售数据的正态分布，可以得到收益的期望。利润与货物的运输方案无直接关系，运输利润与销售数量直接相关，因此本文首先以最大期望利润为目标，确定每种货物运输的数量，再结合优化模型确定装机方案。

5.4.2 模型的建立和求解

（1）确定货物数量的优化模型

由问题 1 种的分析，货物销量数据服从正态分布 $N(\mu_i,\sigma_i)$，以样本均值、样本方差的修正值作为参数的无偏估计，下图为第 i 种货物销售数量的概率分布图：

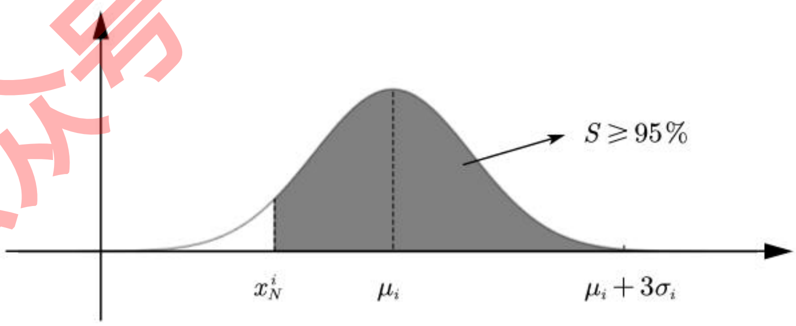

图 10　第 i 种货物销售数量的概率分布图

假设第 i 种货物的供给量（即运送量）为 x_1^i，根据正态分布的 3σ 法则，满足 95%可靠性条件下，分布函数在区间 $(x_1^i,\mu_i+3\sigma_i)$ 内的值小于等于 95%，即图中阴影部分面积满足：

$$\int_{x_1^i}^{\mu_i+3\sigma_i} f_i(x)\times x dx \geq 95\% \tag{26}$$

设第 i 种货物的市场需求为x_N^i，P_N^i为其概率，通过正态分布函数图像，以及数量为整数的限制条件，积分区间长度为 1，计算得：

$$P_N^i = \int_{x_N^i - 0.5}^{x_N^i + 0.5} f_i(x) \times x dx \tag{27}$$

则需求量为x_i^i时的收益为：

$$r_i = \begin{cases} x_N^i \times p_c - x_p \times pf - x_p \times 40\% + (x_p - x_N^i) p_c \times 30\% & (x_p > x_N) \\ x_p \times p_c - x_p \times pf - x_p \times 40\% & (x_p < x_N) \end{cases} \tag{28}$$

其中，pf 为此种货物的运输价格，p_c 为次种货物的售价，当运输量大于需求时$(x_p > x_N)$，收益组成的第 1 项为销售收益，第 2 项为运输成本，第 3 项为货物生产成本，第 4 项为剩余货物甩卖收益，当运输量大于需求时$(x_p < x_N)$，区别在于销售数量为运输数量，且没有剩余货物。

从而得到次运输数量下的期望收益为：

$$R = \sum_{x_N^i \in (\mu^i - 3\sigma_i, \mu^i + 3\sigma_i)} r_i \times P_N^i \tag{29}$$

最终将所有货物的期望收益求和，得到总体期望收益R_{total}。以总体收益的期望为目标函数：$\max(R_{total})$，决策变量为每种货物的运输数量：x_1^i（x_1^i为整数），建立无约束优化模型。

5.4.3 结果分析

表 14 问题四最大期望收益下的运送数量表

	$N_{\max}$	μ_i	$\mu_i - N_{\max}$
HW1 数量	119	119	0
HW2 数量	368	368	0
HW3 数量	361	361	0
HW4 数量	363	364	1
HW5 数量	247	247	0
HW6 数量	307	307	0
HW7 数量	595	611	16
HW8 数量	2947	2993	46
HW9 数量	572	617	45
HW10 数量	720	1225	505

模型计算得到每种货物运输数量在其均值上下3σ范围内的的期望收益，上表中展示了部分结果，将最大期望利润时运送数量$N_{\max}$与前 50 次平均销售数量μ_i，以及两者之差，整理在下表中，其中 HW1 到 HW6 的$N_{\max}$，μ_i几乎相同，即航运公司倾向于多选择

1 到 6 号货物，而 HW7 到 HW10 的 $N_{\max}$ 明显小于 μ_i，航运公司倾向于少选择 7 到 10 号货物，结合方差数据，7 到 10 号货物的方差明显较高，即销量数据波动较大，为满足 95%的可靠性要求，航运公司倾向于选则方差较小，较为稳定的货物 1 到 6，此结果较好地符合了实际情况，最终的总期望利润为 755.27 万元。

5.5 问题五的分析与求解

5.5.1 模型的求解

问题五除可靠性以外的所有条件都与问题四相同，在问题四的模型与算法的基础上，将可靠性 95%改成 70%。因此针对问题五无需建立新的模型求解，直接将 70%的可靠性代入前文建立的优化模型即可解出最大利润值和装运策略。

5.5.2 结果分析

将可靠度降低 25%后，航运公司能够多选择运输销量波动较大的货物 HW7 到 HW10，以可靠性降低为代价，提高期望收益。以下结果为定量分析：

表 15 问题五最大期望收益下的运送数量表

	$N_{\max}$	μ_i	$\mu_i - N_{\max}$
HW1 数量	124	119	0
HW2 数量	368	368	0
HW3 数量	361	361	0
HW4 数量	364	364	0
HW5 数量	247	247	0
HW6 数量	307	307	0
HW7 数量	599	611	12
HW8 数量	2958	2993	35
HW9 数量	573	617	44
HW10 数量	890	1225	335

其中 HW1 到 HW6 $N_{\max}$，μ_i 相同，HW7 到 HW10 的 $N_{\max}$ 依然小于 μ_i，但是相较于问题 4 中 95%的可靠性，在 70%的可靠性条件下 HW7 到 HW10 的运输数量分别有 3.3%，3.5%，2.4%，6.1%的数量增加，当可靠性要求降低后，运输公司有更大的概率选择运输方差较大的货物 HW7 到 HW10。总利润为 793.24 万元，相比问题 4 增加了 4.3%。如果选择三种飞机同时运输每种型号飞机大概需要飞行 75 班次。

六、模型评价

7.1 优点

（1）本文使用的遗传算法具有优秀的全局搜索能力，可通过编码方式对研究对象直接进行操作，可根据实际情况自定义交叉变异概率，较为灵活化。

（2）本文设计的算法和建立的模型适用于解决货物装载策略优化以及货物利润最大化问题。

（3）本文采用了空间分割与空间合并的方法，拓展了剩余空间的大小，有效利用闲置空间，避免空间资源的浪费。

（4）在对货物和集装箱装载进行优化的同时，也直接增加了货物的舱位，从而间接地提高了运输的效率。

（5）此研究对于提高航空货运安全、提高航空货运的经济水平，促使航空运输业态趋于健康发展均具有重要的现实意义。

7.2 缺点

（1）贪心算法总是从局部出发，没从整体考虑，不能保证解是最佳的。

（2）遗传算法整体运行速度慢，容易早熟。

（3）本文的模型和算法适用于规则形状集装箱，但实际集装箱一般贴合于飞机底部形状，多为不规则形状，日后需改进模型对不规则形状集装箱装载进行优化。

（4）本文对定性影响因素的研究考虑不足。

7.3 推广

货运飞机的飞行成本高，货舱舱资源有限，因而货舱的载货利用率直接影响到每架次飞行的综合效益。另外，机型变化，集装器型号不同，装载作业时间有限，货运规划的规划效率以及规划的科学水平、工作人员的作业效率等都会影响到货舱舱位的利用水平。使用货舱装载货物时，对于货物有相关的种类、尺寸等限制，所以航空货运资源的多机分配与优化问题也具有重要的意义。

七、参考文献

[1]韩明. 概率论与数理统计. 上海. 同济大学出版社. 2019(08):92-108
[2]王彩霞. 考虑双边可靠性的物流服务供应链协调契约研究[J]. 中国集体经济,2020,(26):103-104.
[3]史永胜,王策. 货机主货舱多约束条件下集装器装载优化[J]. 科学技术与工程,2020,20(25):10517-10522.
[4]孟超. 民航宽体客机腹舱装载优化研究[D].中国民航大学,2020.
[5]周鑫. 民用货机货物装载系统维修性研究[D].中国民航大学,2020.
[6]王策. 货机货物装载进程可视化系统的设计与实现[D].中国民航大学,2020.
[7]胡京煜. 基于模型的系统工程在无人货运飞机概念设计中的应用[D].南京航空航天大学,2020.
[8]连晓东,方永红,陈莉,余莹屏. 关于支线集装箱船货舱装载多种箱型的研究[J]. 广东造船,2020,39(01):102-104.
[9]冯成骁,王宗军,钟琴,谭洁. 基于供应商可靠性成本投入的零售商激励策略问题[J]. 管理学报,2019,16(11):1729-1738.
[10]罗武林,刘杰,张中. 基于有遗传算法的货运飞机智能配载系统开发[J]. 科技创新导报,2019,16(01):8+10.
[11]贾旭颖. 民航货机装载优化模型的构建与实现[D].中国民航大学,2018.
[12]谢万祯,杨睿颀,陈灏,夏延龙,郭晶晶,卢飞. 基于有遗传算法的货运飞机智能配载系统开发[J]. 电脑迷,2018,(04):130-132.
[13]孙利,王德禹. 支持向量机和遗传算法组合策略的 VLCC 船中结构优化设计(英文)[J]. Journal of Marine Science and Application,2012,11(01):59-67.
[14]王高阳,夏利娟. 基于组合算法的 VLCC 货舱区综合分舱优化[J]. 中国造船,2020,61(04):198-208.
[15]谷润平，刘申申，胡皓. 最大货舱可用业载的机队调配方法研究［J］. 航空计算技术，2017，47(3) : 9-12.
[16]王鹏，王和平. 基于多目标遗传算法的大型客机总体参数优化［J］. 航空计算技术，2012，42(3) : 30-32，37.
[17]杨广全，梁永刚，刘平. 基于遗传算法的铁路小型集装箱装车配载优化［J］. 铁道货运，2017(5) : 41-45.
[18]周柯雯,姚爱祥.基于遗传算法和启发策略的军用物资飞机装载优化[J].物流科技,2010,33(09):122-124.
[19]谢万祯,杨睿颀,陈灏,等.基于有遗传算法的货运飞机智能配载系统开发[J].电脑迷,2018(10):130-132.
[20]Yanzhi Li, Yi Tao, Fan Wang. A compromised large-scale neighborhood search heuristic for capacitated air cargo loading planning[J]. European Journal of Operational Research, 2009
[21]S.G. Jin,O. Luo,C. Ren. Effects of physical correlations on long-distance GPS positioning and zenith tropospheric delay estimates[J]. Advances in Space Research,2010,46(2)
[22]Chunhua Tian, Hao Zhang, Feng Li, Tie Liu. Air cargo load planning system:Arule-based optimization approach[C]. Chicago:IEEE International Conference on

八、附录

附录一：主要程序

附录 1. 1

问题一求解代码（lingo）

```
model:

sets:
kind/1..10/:k1,hy,v;!货物种类;
fj/1..3/;!飞机种类;
hc/1..3/;!货舱种类;
link0(hc,fj):w1;!货物限重;
link1(hc,fj,kind):n1;

endsets

data:
k1=2.1 0.2 0.7 1.8 1.3 0.3 0.23 1.2 0.9 0.3;
v=7.592992 1.157625 5.71234 5.0673 2.40597 0.709632 0.214032 2.47 2.8704 1.5;
w1=10 8 6
   16 12 8
   8 6 4;
hy=119 368 361 364 247 307 611 2993 617 1225;
!@text('问题的解.txt')=y;
enddata

max = @ sum(kind(k):@ sum(fj(n):@ sum(hc(m):n1(m,n,k)))*v(k));

@ for(fj(n):@ for(hc(m):@ sum(kind(k):n1(m,n,k)*k1(k))<=w1(m,n)));
@ for(fj(n):@ sum(kind(k): n1(1,n,k)*k1(k))/@ sum(kind(k): n1(2,n,k)*k1(k))/@
sum(kind(k):n1(3,n,k)*k1(k))=w1(1,n)/w1(2,n)/w1(3,n));
@ for(kind(k):@ sum(fj(n):@ sum(hc(m):n1(m,n,k)))<=hy(k));

@ for(link1(m,n,k):@ gin(n1(m,n,k)));
End
```

附录 1.2

问题一求解代码（lingo）

```
model:

sets:
kind/1..10/:k1,hy;!货物种类;
fj/1..3/;!飞机种类;
hc/1..3/;!货舱种类;
link0(hc,fj):w1;!货物限重;
t1/1..78/;!大飞机的班次;
t2/1..78/;!中飞机的班次;
t3/1..78/;!小飞机的班次;
link1(t1,kind,hc):n1;!大飞机每个班次、每个货舱、每种材料的货物数量;
link2(t2,kind,hc):n2;
link3(t3,kind,hc):n3;

endsets

data:
k1=2.1 0.2 0.7 1.8 1.3 0.3 0.23 1.2 0.9 0.3;!每种货物每件质量;
w1=10 8 6
   16 12 8
   8 6 4;
hy=107 368 361 364 247 307 0 2967 605 1199;!每种材料货源量;

enddata

min = 84*6242.86+84*2851.244+84*363.63-17180.375954;

!大飞机;
@ for(t1(t):@ for(hc(m):@ sum(kind(k):n1(t,k,m)*k1(k))=w1(m,1)));!最大载重限制;
!@ for(t1(t):@ sum(kind(k):n1(t,k,1)*k1(k))/@ sum(kind(k):n1(t,k,2)*k1(k))/@
sum(kind(k):n1(t,k,3)*k1(k))=10/16/8);!重量比例;
@ for(t1(t):@ sum(kind(k):n1(t,k,1)*k1(k))=10);!重量比例;
@ for(t1(t):@ sum(kind(k):n1(t,k,2)*k1(k))=16);!重量比例;
@ for(t1(t):@ sum(kind(k):n1(t,k,3)*k1(k))=8);!重量比例;
!@ for(t1(t):@ sum(kind(k):n1(t,k,1)*k1(k))/@ sum(kind(k):n1(t,k,2)*k1(k))/@
sum(kind(k):n1(t,k,3)*k1(k))<=10/16/8*1.05);!重量比例;
!@ for(t1(t):@ sum(kind(k):n1(t,k,1)*k1(k))/@ sum(kind(k):n1(t,k,2)*k1(k))/@
sum(kind(k):n1(t,k,3)*k1(k))>=10/16/8*0.95);!重量比例;
```

```
!中飞机;
@ for(t2(t):@ for(hc(m):@ sum(kind(k):n2(t,k,m)*k1(k))=w1(m,2)));!最大载重限制;
!@ for(t2(t):@ sum(kind(k):n2(t,k,1)*k1(k))/@ sum(kind(k):n2(t,k,2)*k1(k))/@
sum(kind(k):n2(t,k,3)*k1(k))=8/12/6);!重量比例;
@ for(t2(t):@ sum(kind(k):n2(t,k,1)*k1(k))=8);!重量比例;
@ for(t2(t):@ sum(kind(k):n2(t,k,2)*k1(k))=12);!重量比例;
@ for(t2(t):@ sum(kind(k):n2(t,k,3)*k1(k))=6);!重量比例;
!@ for(t2(t):@ sum(kind(k):n2(t,k,1)*k1(k))/@ sum(kind(k):n2(t,k,2)*k1(k))/@
sum(kind(k):n2(t,k,3)*k1(k))<=8/12/6*1.05);!重量比例;
!@ for(t2(t):@ sum(kind(k):n2(t,k,1)*k1(k))/@ sum(kind(k):n2(t,k,2)*k1(k))/@
sum(kind(k):n2(t,k,3)*k1(k))>=8/12/6*0.95);!重量比例;


!小飞机;
@ for(t3(t):@ for(hc(m):@ sum(kind(k):n3(t,k,m)*k1(k))=w1(m,3)));!最大载重限制;
@ for(t3(t):@ sum(kind(k):n3(t,k,1)*k1(k))=6);!重量比例;
@ for(t3(t):@ sum(kind(k):n3(t,k,2)*k1(k))=8);!重量比例;
@ for(t3(t):@ sum(kind(k):n3(t,k,3)*k1(k))=4);!重量比例;
!@ for(t3(t):@ sum(kind(k):n3(t,k,1)*k1(k))/@ sum(kind(k):n3(t,k,2)*k1(k))/@
sum(kind(k):n3(t,k,3)*k1(k))<=6/8/4*1.05);!重量比例;
!@ for(t3(t):@ sum(kind(k):n3(t,k,1)*k1(k))/@ sum(kind(k):n3(t,k,2)*k1(k))/@
sum(kind(k):n3(t,k,3)*k1(k))>=6/8/4*0.95);!重量比例;

!@ for(kind(k): (@ sum(link5:(n1(t,k,m)))+@ sum (link6:(n1(t,k,m)))+@ sum (link7:(n1(t,k,m))))=hy(k));

!货源限制;
@ for(kind(k): ((@ sum (t1(t):@ sum(hc(m): n1(t,k,m)))) + (@ sum (t2(t):@ sum(hc(m): n2(t,k,m)))) +(@ sum
(t3(t):@ sum(hc(m): n3(t,k,m)))))=hy(k));!货源数量限制;

@ for(link1(t,k,m):@ gin(n1(t,k,m)));
@ for(link2(t,k,m):@ gin(n2(t,k,m)));
@ for(link3(t,k,m):@ gin(n3(t,k,m)));
End
```

附录 1.3

问题二求解代码（python）（遗传算法）

```
# import numpy as np
import random
import math
import datetime
import pandas as pd
from main_copy import pack_products_into_restrictions

item_sets = [[1.05, 1.05, 1.05], [1.5, 1, 1], [1.32, 0.64, 0.84], [1.98, 0.42, 0.53]]
Box = (2.891, 2.338, 2.338)

item_sets.sort(key=lambda x: (x[0] * x[1] * x[2]), reverse=True)  # 将箱子按从大到小排列
item_sets = [sorted(i, reverse=True) for i in item_sets]  # 将箱子的摆放顺序统一

def exchange_item(items):  # 第一类邻域选择，随机交换两个物品
    s1, s2 = random.randint(0, len(items) - 1), random.randint(0, len(items) - 1)
    while s1 == s2:
        s2 = random.randint(0, len(item_sets) - 1)
    items[s1], items[s2], = items[s2], items[s1]
    return items

def exchange_direction(items):  # 第二类邻域选择，随机交换某个物品的方向
    s = random.randint(0, len(items) - 1)
    item = items[s]
    s_1, s_2 = random.randint(0, len(item) - 1), random.randint(0, len(item) - 1)
    while s_1 == s_2:
        s_2 = random.randint(0, len(item) - 1)
    item[s_1], item[s_2], = item[s_2], item[s_1]
    items[s] = item
```

```
    return items
def sa(alpha, t_set, items_sa, box_sa, markovlen):
    # alpha = 0.99
    # t = (1, 100)
    # m = 100
    min_t = t_set[0]
    t = t_set[1]
    valuecurrent = pack_products_into_restrictions(items_sa,box_sa)[0]
    valuebest = valuecurrent
    itemscurrent = items_sa
    result = []   # 记录迭代过程中的最优解
    while t > min_t:
        for i in range(markovlen):
            # 倒序+插段
            if random.random() > 0.5:   # 交换路径中的这 2 个节点的顺序
                itemsnew = exchange_item(items_sa)
            else:   # 交换次序
                itemsnew = exchange_direction(items_sa)

            valuenew, select_items = pack_products_into_restrictions(items_sa,box_sa)
            # print (valuenew)
            if valuenew >= valuecurrent:   # 接受该解
                r = 0
                # 更新 solutioncurrent 和 solutionbest
                valuecurrent = valuenew
                itemscurrent = itemsnew.copy()
                if valuenew >= valuebest:
                    valuebest = valuenew
                    itemsbest = select_items.copy()
            else:   # 按一定的概率接受该解
                if random.random() <= math.exp(-(valuecurrent - valuenew) / t):
```

```
                    # if np.random.rand() < (2/math.pi) * math.atan((valuenew - valuecurrent) * 0.000001*t):
                        valuecurrent = valuenew
                        itemscurrent = itemscurrent.copy()
                    else:
                        itemsnew = itemscurrent.copy()
                t = alpha * t
            # result.append(itemsbest)
                print('temp:', t)
                print('itemsbest', itemsbest)
                print('valuebest', valuebest)
s1 = datetime.datetime.now()
sa(0.90, (0.01, 1), [[8, 8, 9], [7, 5, 8], [7, 4, 2], [1, 2, 1], [2, 2, 1], [1, 1, 3], [4, 2, 2], [2, 2, 1], \
                        [1, 1, 3], [4, 2, 2], [2, 2, 1], [1, 1, 3], [4, 2, 2], [4, 4, 4], [3, 3, 3]], (10, 10, 10), 1)
s2 = datetime.datetime.now()
print('time:', s2 - s1)
#   pack_products_into_restrictions(item_sets, Box)
```

附录 1. 4

```
clc
clearvars -except dataC
C=table2array(dataC);
Ca=table2array(dataC(:,1:10));
data1=C(find(C(:,2)==1),:);
data2=C(find(C(:,2)==2),:);
data3=C(find(C(:,2)==3),:);
data1a=Ca(find(Ca(:,2)==1),:);
data2a=Ca(find(Ca(:,2)==2),:);
data3a=Ca(find(Ca(:,2)==3),:);
for i=1:size(data1a,1)
%for i=1:1
    b=sort(data1a(i,3:size(data1a,2)))
    for j=1:size(data1a(i,3:size(data1a,2)),2)
    data1ado(i,j)=find(b(1,j)==data1a(i,3:size(data1a,2)));
    end
```

```
end
%%
plot(1,data1(:,3),'*')
hold on
plot(1,data2(:,3),'o')
hold on
plot(1,data3(:,3),'.')
hold on
```

附录 1.5

```
clc
clear all
%%
a= ...
[120    119    117    120    122    119    119    120    120    119
 119    120    119    121    120    121    119    118    119    118
 120    120    121    118    121    119    118    121    118    119
 119    121    118    119    118    120    120    120    119    117
 120    119    120    118    119    117    120    119    119    120]
b(:,1)=a(:)
a= ...
[369    367    367    368    365    368    365    370    367    369
 367    366    366    365    363    367    368    365    367    371
 367    370    365    366    365    368    366    370    367    368
 366    367    373    367    368    367    367    369    368    367
 370    366    371    369    368    369    369    367    368    369]
b(:,2)=a(:)
a= ...
[362    363    360    358    358    360    363    361    358    357
 358    362    367    361    366    361    358    362    364    361
 364    359    359    361    365    362    359    352    363    366
 362    361    360    366    361    361    359    360    361    362
 362    361    359    361    357    363    360    365    362    362]
b(:,3)=a(:)
a= ...
[363    364    367    361    365    363    366    365    364    364
 364    369    361    365    367    362    366    363    361    360
 365    360    363    362    367    366    365    363    367    366
 365    369    364    365    363    365    367    367    366    360
 364    365    365    366    365    364    366    361    365    360]
```

```
b(:,4)=a(:)
a= ...
[248    246    249    246    245    248    246    246    245    251
 246    248    247    245    247    246    245    252    247    247
 248    248    247    247    248    247    247    250    248    248
 249    249    250    249    247    243    245    243    249    249
 249    249    243    249    248    250    247    248    247    248]
b(:,5)=a(:)
a= ...
[309    306    307    308    305    306    312    311    306    307
 308    308    307    307    308    307    309    309    307    304
 306    309    310    305    306    308    307    309    306    309
 304    308    306    307    307    304    308    306    305    307
 311    306    309    307    308    310    305    308    306    309]
b(:,6)=a(:)
a= ...
[556    546    558    664    702    476    641    518    531    688
 559    697    556    563    745    380    542    651    533    618
 698    547    800    670    604    566    614    598    686    655
 570    617    695    741    774    671    436    484    611    668
 714    697    556    439    615    631    523    599    639    717]
 b(:,7)=a(:)
a= ...
[2851          2837          3202          3031          3064          3106          2924          2959          3007
2861
 2995          3016          2764          3013          2957          3115          2943          2983
3132          3031
 3096          3037          2949          2928          3104          3005          3055          3040
2978          2975
 3173          2977          2867          3077          3066          2871          2944          2904
2986          3050
 2956          2885          2936          3062          3250          2962          2910          3031
2882          2910]
b(:,8)=a(:)
a= ...
[840    719    508    727    691    598    632    507    592    653
 662    705    655    693    684    560    529    548    624    735
 754    537    608    641    684    529    620    628    493    690
 627    496    613    752    577    753    326    617    602    579
 633    540    513    562    682    565    541    844    516    486]
b(:,9)=a(:)
a= ...
[1492          1081          1284          1394          961          889          1410          1757          915
```

```
1280
 1610        1106        1280        1086        1329        1234        1050        1345
1343        1385
 1224        1377        1219        1361        1129        1187        1012        1045
1044        878
 1002         922        1299        1234        1209        1439         946        1367
1263        1332
 1439         808        1416        1098        1041        1360        1299         974
1481        1627]
b(:,10)=a(:)
%%
tiaojian=round(mean(b))
```

附录 1.6

```
clc
clear all
%%
a= ...
[120   119   117   120   122   119   119   120   120   119
 119   120   119   121   120   121   119   118   119   118
 120   120   121   118   121   119   118   121   118   119
 119   121   118   119   118   120   120   120   119   117
 120   119   120   118   119   117   120   119   119   120]
b(:,1)=a(:)
a= ...
[369   367   367   368   365   368   365   370   367   369
 367   366   366   365   363   367   368   365   367   371
 367   370   365   366   365   368   366   370   367   368
 366   367   373   367   368   367   367   369   368   367
 370   366   371   369   368   369   369   367   368   369]
b(:,2)=a(:)
a= ...
[362   363   360   358   358   360   363   361   358   357
 358   362   367   361   366   361   358   362   364   361
 364   359   359   361   365   362   359   352   363   366
 362   361   360   366   361   361   359   360   361   362
 362   361   359   361   357   363   360   365   362   362]
```

```
b(:,3)=a(:)
a= ...
[363    364    367    361    365    363    366    365    364    364
 364    369    361    365    367    362    366    363    361    360
 365    360    363    362    367    366    365    363    367    366
 365    369    364    365    363    365    367    367    366    360
 364    365    365    366    365    364    366    361    365    360]
b(:,4)=a(:)
a= ...
[248    246    249    246    245    248    246    246    245    251
 246    248    247    245    247    246    245    252    247    247
 248    248    247    247    248    247    247    250    248    248
 249    249    250    249    247    243    245    243    249    249
 249    249    243    249    248    250    247    248    247    248]
b(:,5)=a(:)
a= ...
[309    306    307    308    305    306    312    311    306    307
 308    308    307    307    308    307    309    309    307    304
 306    309    310    305    306    308    307    309    306    309
 304    308    306    307    307    304    308    306    305    307
 311    306    309    307    308    310    305    308    306    309]
b(:,6)=a(:)
a= ...
[556    546    558    664    702    476    641    518    531    688
 559    697    556    563    745    380    542    651    533    618
 698    547    800    670    604    566    614    598    686    655
 570    617    695    741    774    671    436    484    611    668
 714    697    556    439    615    631    523    599    639    717]
 b(:,7)=a(:)
a= ...
[2851          2837          3202          3031          3064          3106          2924
2959          3007          2861
 2995          3016          2764          3013          2957          3115          2943
2983          3132          3031
 3096          3037          2949          2928          3104          3005          3055
3040          2978          2975
 3173          2977          2867          3077          3066          2871          2944
2904          2986          3050
 2956          2885          2936          3062          3250          2962          2910
3031          2882          2910]
b(:,8)=a(:)
a= ...
[840    719    508    727    691    598    632    507    592    653
```

```
 662     705     655     693     684     560     529     548     624     735
 754     537     608     641     684     529     620     628     493     690
 627     496     613     752     577     753     326     617     602     579
 633     540     513     562     682     565     541     844     516     486]
b(:,9)=a(:)
a= ...
[1492           1081           1284           1394            961            889           1410
1757            915            1280
 1610           1106           1280           1086           1329           1234           1050
1345           1343           1385
 1224           1377           1219           1361           1129           1187           1012
1045           1044           878
 1002            922           1299           1234           1209           1439            946
1367           1263           1332
 1439            808           1416           1098           1041           1360           1299
974           1481           1627]
b(:,10)=a(:)
%%
tiaojian=round(mean(b))
%%
huowu= ...
[2.96     2.12      1.21      2.1
 1.05     1.05      1.05      0.2
 3.17     0.85      2.12      0.7
 5.08     0.95      1.05      1.8
 2.01     1.26      0.95      1.3
 1.32     0.64      0.84      0.3
 0.98     0.42      0.52      0.23
 2         1.3       0.95      1.2
 2.08     1.2       1.15      0.9
 1.5      1         1         0.3]
huocang1= ...
[18 8.45 13.4 10
 20 8.45 14.8 16
 16 8.45 12.6 8]
huocang2= ...
[15 5.48 10.2 8
 18 5.48 13.4 12
 13 5.48 9.7 6]
huocang3= ...
[10 3.45 3.4 6
 12 3.45 3.4 8
 9   3.45 3.4 4]
```

```
for i=1:10
hwmd(i,:)=huowu(i,4)/(huowu(i,1)*huowu(i,2)*huowu(i,3))
end
%%
% k = 10;                                    % 组合阶数
% n = 10^k;                                  % 组合数量
% index = 0:n-1;                             % 组合序号
% index = index';                            % 转置
% nBin = index;                              % 十进制转换二进制
% C2=nBin+1
%%
huowu(:,5)=hwmd
huowu(:,6)=tiaojian'
huowuzhongliang=sort(huowu(:,4),'descend')
%%
huowu=sortrows(huowu,5)
pda=[]
for i=1:10
pdmubiao=[]
pdmubiao(1:huowu(i,6),1)=huowu(i,4)
pda=[pda;pdmubiao]
end
size(find(pda(:,1)==0.7),1)
size(find(pda(:,1)==0.2),1)
size(find(pda(:,1)==0.3),1)
size(find(pda(:,1)==2.1),1)
size(find(pda(:,1)==0.9),1)
size(find(pda(:,1)==1.8),1)
size(find(pda(:,1)==1.2),1)
size(find(pda(:,1)==1.3),1)
size(find(pda(:,1)==0.23),1)
```

附录 1.7

```
chang=18
kuan=8.45
gao=13.4
pailie=perms(1:3)
for j=3
for i=1:size(pailie,1)
onesways(i,1)=fix(chang/huowu(j,pailie(i,1)))
onesways(i,2)=fix(kuan/huowu(j,pailie(i,2)))
onesways(i,3)=fix(gao/huowu(j,pailie(i,3)))
onesways(i,4)=onesways(i,1)*onesways(i,2)*onesways(i,3)
onesways(i,5)=chang-fix(chang/huowu(j,pailie(i,1)))*huowu(j,pailie(i,1))
onesways(i,6)=kuan-fix(kuan/huowu(j,pailie(i,2)))*huowu(j,pailie(i,2))
onesways(i,7)=gao-fix(gao/huowu(j,pailie(i,3)))*huowu(j,pailie(i,3))
onesways(i,8)=chang*kuan*gao-huowu(j,1)*huowu(j,2)*huowu(j,3)*onesways(i,4)
end
end
```

附录 1.8

```
%%
hcz=[huocang1(:,4);huocang2(:,4);huocang3(:,4)]
jl=1,j=1,jishu=1,what=1,what2=1
for cs=1:10000
    %%
    jll=jl
    for ii=jl:7212
    pda(ii,2)=sum(pda(jl:ii));
    pda(ii,3)=size(pda(jl:ii,:),1);
        if pda(ii,2)>=hcz(j)
            pda(ii,2)=0;
            pda(ii,3)=0;
            break;
        end
    end
    for css=1:20
        min(pda(ii:7212,1))
```

```
            finm=find(pda(ii+1:7212,1)==min(pda(ii+1:7212,1)));
            pda=[pda(1:ii-1,:);[min(pda(ii+1:7212,1)),0,0];pda(ii:7212,:)];
            pda((ii+finm(1)-1),:)
            if   1
                pda((ii+finm(1)-1),:)=[];
                what=what+1;
            end
            what2=what2+1;
            pda(ii,2)=sum(pda(jll:ii));
            pda(ii,3)=size(pda(jll:ii,:),1);
            ii=ii+1;
            if pda(ii-1,2)-hcz(j)>0.005
                pda=[pda(1:ii-2,:);pda(ii:7212,:);pda(ii-1,:)];
                jl=ii-1;
                break;
            end
        end
        j=j+1
        if j==10
            j=1
        end
        if ii==7212
            break;
        end
        %%
end
%%
sum(pda(:,1))/sum(hcz)
sum(pda(:,1))/sum(hcz)*3
%%
size(find(pda(:,3)==1),1)/9
size(find(pda(:,1)==0.7),1)
size(find(pda(:,1)==0.2),1)
size(find(pda(:,1)==0.3),1)
size(find(pda(:,1)==2.1),1)
size(find(pda(:,1)==0.9),1)
size(find(pda(:,1)==1.8),1)
size(find(pda(:,1)==1.2),1)
size(find(pda(:,1)==1.3),1)
size(find(pda(:,1)==0.23),1)
```

附录 1.9

```
for i=1:10
[h1(i,:),p1(i,:),lstat1(i,:),cv1(i,:)]=lillietest(b(:,i),0.05)
[h2(i,:),p2(i,:),jbstat2(i,:),cv2(i,:)]=jbtest(b(:,i),0.05)
h3(i,:)=adtest(b(:,i))
%figure(2*i-1)
%hist(b(:,i))
%hold on
%figure(2*i)
%normplot(b(:,i))
[muhat(:,i),sigmahat(:,i),muci(:,i),sigmaci(:,i)]= normfit(b(:,i),0.05)
%x=muhat-3*sigmahat:.1:muhat+3*sigmahat;
%norm=gaussmf(x,[sigmahat,muhat]);
%figure(2*i-1)
%plot(x,norm)
end

kank=[roundn(muhat,0);tiaojian]
```


附录 1.10

```
clear hh
hh=huowu(:,4)
for i=1:100
hh(:,i+1)=hh(:,i)+hh(:,1)
end
%%
al=[al4;al6;al8;al10;al12;al16]
%%
all()
```

附录 1. 11

```
jzx=[2.991 2.438 2.438 3 0.67
     2.991 2.438 2.438 1 0.3
     1.456 2.438 1.9    3 0.15
     2.991 2.438 2.438 1 0.15
     2.991 2.438 2.438 2 0.28
     1.46   2.438 1.9    1 0.17
     2.235 2.743 2       1 0.15]

cang=[huocang1(1:3,1:3);huocang2(1:3,1:3);huocang3(1:3,1:3)]

for k=1:9
chang=cang(k,1)
kuan=cang(k,2)
gao=cang(k,3)
pailie=perms(1:3)
for j=1:7
for i=1:size(pailie,1)
oneswa(i,1)=fix(chang/jzx(j,pailie(i,1)))
oneswa(i,2)=fix(kuan/jzx(j,pailie(i,2)))
oneswa(i,3)=fix(gao/jzx(j,pailie(i,3)))
oneswa(i,4)=oneswa(i,1)*oneswa(i,2)*oneswa(i,3)
oneswa(i,5)=chang-fix(chang/jzx(j,pailie(i,1)))*jzx(j,pailie(i,1))
oneswa(i,6)=kuan-fix(kuan/jzx(j,pailie(i,2)))*jzx(j,pailie(i,2))
oneswa(i,7)=gao-fix(gao/jzx(j,pailie(i,3)))*jzx(j,pailie(i,3))
oneswa(i,8)=chang*kuan*gao-jzx(j,1)*jzx(j,2)*jzx(j,3)*oneswa(i,4)
end
janss(j,k)=min(oneswa(:,4))
janss2(j,k)=max(oneswa(:,4))
end
end
unique(janss2,'row')
```

附录 1.12

```
for k=1:7
chang=jzx(k,1)-2*0.05
kuan=jzx(k,2)-2*0.05
gao=jzx(k,3)-2*0.05
pailie=perms(1:3)
for j=1:10
for i=1:size(pailie,1)
onways(i,1)=fix(chang/huowu(j,pailie(i,1)))
onways(i,2)=fix(kuan/huowu(j,pailie(i,2)))
onways(i,3)=fix(gao/huowu(j,pailie(i,3)))
onways(i,4)=onways(i,1)*onways(i,2)*onways(i,3)
onways(i,5)=chang-fix(chang/huowu(j,pailie(i,1)))*huowu(j,pailie(i,1))
onways(i,6)=kuan-fix(kuan/huowu(j,pailie(i,2)))*huowu(j,pailie(i,2))
onways(i,7)=gao-fix(gao/huowu(j,pailie(i,3)))*huowu(j,pailie(i,3))
onways(i,8)=chang*kuan*gao-huowu(j,1)*huowu(j,2)*huowu(j,3)*onways(i,4)
end
ccjanss(j,k)=min(onways(:,4))
ccjanss2(j,k)=max(onways(:,4))
end
end

huowu(:,7)=huowu(:,1).*huowu(:,2).*huowu(:,3)
ccjanss2=ccjanss2(find(huowu(:,7)<2.01),:)

unique(ccjanss2','row')'
```

附录 1.13

```
clearvars -except tiaojian sigmahat sigmaci muci
tj=tiaojian'
zl=[2.1;0.2;0.7;1.8;1.3;0.3;0.23;1.2;0.9;0.3]
jgg= ...
[2800 8000;450 1500;2400 8500;3100 8800; ...
1200 4200;280 1200;150 500;1000 3800;900 3500;500 2000]
jgg(:,3)=0.4*jgg(:,2)
jgg(:,4)=0.3*jgg(:,2)
cb=jgg(:,1)+jgg(:,3)
```

```
zlirun=jgg(:,2)-cb
clirun=jgg(:,4)-cb
maishenme=zlirun+clirun
%%
for ij=1:10
zjia(ij,1)=0
suans=0
for xx=floor(tj(ij,1)-4*sigmahat(1,ij)):ceil(tj(ij,1)+4*sigmahat(1,ij))
zt=normpdf(xx,tj(ij,1),sigmahat(1,ij))
lzzz=normpdf(119,119,1.3)
suans=suans+zt
if xx<tj(ij,1)
    1
    zjia(ij,1)=zt*((tj(ij,1)-xx)*clirun(ij,:)+xx*zlirun(ij,:))+zjia(ij,1)
else
    zjia(ij,1)=zt*tj(ij,1)*zlirun(ij,:)+zjia(ij,1)
    2
end
%zjia(ij,1)=zjia(ij,1)/tj(ij,1)
end
end
%%
for ij=1:10
zjia(ij,2)=0
for xx=floor(tj(ij,1)-4*sigmaci(1,ij)):ceil(tj(ij,1)+4*sigmaci(1,ij))
zt=normpdf(xx,tj(ij,1),sigmaci(1,ij))
if xx<tj(ij,1)
    zjia(ij,2)=zt*((tj(ij,1)-xx)*clirun(ij,:)+xx*zlirun(ij,:))+zjia(ij,2)
else
    zjia(ij,2)=zt*tj(ij,1)*zlirun(ij,:)+zjia(ij,2)
end
%zjia(ij,2)=zjia(ij,2)/tj(ij,1)
end
end
%%
for ij=1:10
suans=0
zjia(ij,3)=0
for xx=floor(tj(ij,1)-4*sigmaci(2,ij)):ceil(tj(ij,1)+4*sigmaci(2,ij))
zt=normpdf(xx,tj(ij,1),sigmaci(2,ij))
suans=suans+zt
if xx<tj(ij,1)
    zjia(ij,3)=zt.*((tj(ij,1)-xx).*clirun(ij,:)+xx.*zlirun(ij,:))+zjia(ij,3)
```

```
else
	zjia(ij,3)=zt*tj(ij,1)*zlirun(ij,:)+zjia(ij,3)
end
%zjia(ij,3)=zjia(ij,3)/tj(ij,1)
end
end
%%
zjia=roundn(zjia,0)
%%
zjia(:,1)./zl
%%
zlirun.*tj
```

附录 1.14

```
yunme=jgg(:,4)-jgg(:,1) %运过去赚的钱
jgg(:,2)-jgg(:,3)-jgg(:,1)
fahuo=[],zt9=[],zt99=[]
%%
for shangpin=4
	ij=shangpin
	ifh=1
%for xgh=floor(tj(ij,1)-4*sigmahat(1,ij)):ceil(tj(ij,1)+4*sigmahat(1,ij))
for xgh=364
for xxy=floor(tj(ij,1)-4*sigmahat(1,ij)):ceil(tj(ij,1)+4*sigmahat(1,ij))
if yunme(shangpin,1)>0
	cbenen=xgh*(jgg(shangpin,3)+jgg(shangpin,1));
	if xxy>=xgh
		xiaoshou=xgh*jgg(shangpin,2);
	else
		xiaoshou=xxy*jgg(shangpin,2)+(xgh-xxy)*jgg(shangpin,4);
	end
	dqklr(xgh,xxy)=xiaoshou-cbenen;
else
	if xxy>=xgh
		cbenen=xgh*jgg(shangpin,3)+xgh*jgg(shangpin,1);
		fahuo(ifh)=xgh;
		ifh=ifh+1;
	else
		cbenen=xgh*jgg(shangpin,3)+xxy*jgg(shangpin,1);
		fahuo(ifh)=xxy;
```

```
            ifh=ifh+1;
        end
        if xxy>=xgh
            xiaoshou=xgh*jgg(shangpin,2);
        else
            xiaoshou=xxy*jgg(shangpin,2)
        end
        dqklr(xgh,xxy)=xiaoshou-cbenen;
end
zt9(ifh)=normpdf(xxy,tj(ij,1),sigmahat(1,ij))
end
end
end
dqklr(find(max(dqklr)==dqklr & dqklr~=0))';
zt99=zt9(1,2:size(zt9,2))
sum(zt99)
sum(fahuo.*zt99)
```

附录二：主要结果

附表 2.1

附表 2.2

附表 2.3