

基于规划模型的军事补给问题研究

摘要

现代战争中，士兵往往不会一次性带上大量的食物和弹药，如果陷入了较为持久的战斗状态，就需要调动补给站中的后勤部队快速相应，补充物资。本文综合考虑各种因素，建立了选取最优调度方案的数学模型。

针对问题一，要确定新补给站的修建地点，且仅仅考虑最大程度给各个部队提供支援，那么实际上判断条件为距各点的距离。本文以 C++ 为工具，使用最小圆覆盖算法和模拟退火算法建立模型，求出满足坐标系中各部队都在内的以新补给站为圆心的最小的圆的圆心半径。最终求得新补给站的修建地点为(10.5,4.0)。

针对问题二，题目中已经给出了新补给站的位置，需要确定飞机的调用方案。首先，我们通过合理的方式简化了问题；然后我们题目以物资转运量最大为目标，以飞机调往四个部队的两种类型飞机数量为决策变量，以四个部队的物资需求量和最短需求时间为限制，通过改变约束条件，目的是优化模型，建立了多种线性整数规划模型；最后，我们得到了三种飞机调用方案；最后，为了选择最优方案，我们以熵权 TOPSIS 组合模型为评价模型，通过评价模型计算三个方案的得分，计算得到第三种方案为最优方案，最后我们选择的飞机调用方案是：首先，飞往 A101 部队，A 类飞机 1 架，B 类飞机 0 架；飞往 A122 部队，A 类飞机 3 架，B 类飞机 2 架；飞往 B121 部队，A 类飞机 1 架，B 类飞机 4 架；飞往 B404 部队，A 类飞机 1 架，B 类飞机 4 架。然后，当 A 类飞机返回后，立即进行第二次转运，将 6 架 A 类飞机全部飞往 A101 部队。总耗时 101 分钟，总运量 356 吨。

针对问题三，题目限制部队行进不能携带超过所需数量 110% 的物资，且飞机会在转运途中有少量的物资损耗。问题三沿用问题二中得分最高的数学模型，并使用遗传算法进行优化。由于物资损耗率为不可控数据，我们将损耗率的下限和上限作为转运可接受风险的上限和下限，分别计算损耗率最低和损耗率最高的最优调用方案。此数学模型需要决策者考虑可接受的风险程度来选取方案，当可接受风险较高时，应当选择损耗率较低时的调用方案；如果可接受风险较低时，则选择损耗率较高时的调用方案。

关键词：整数规划、模拟退火算法、最小覆盖圆算法、熵权 TOPSIS 组合模型、遗传算法

目录

一、问题重述.....	3
1.1 问题背景.....	3
1.2 已知条件.....	3
1.3 解决问题.....	3
二、问题分析.....	4
2.1 问题一分析.....	4
2.2 问题二分析.....	4
2.3 问题三分析.....	5
三、模型假设.....	5
四、定义与符号说明.....	5
五、模型的创立与求解.....	5
5.1 准备工作.....	5
5.1.1 数据可视化.....	5
5.1.2 数据处理.....	6
5.2 问题一的模型.....	7
5.2.1 最小圆覆盖算法.....	7
5.2.2 模拟退火算法.....	7
5.2.3 求解.....	8
5.3 问题二的模型.....	8
5.3.1 模型 I.....	8
5.3.2 模型 II.....	10
5.3.3 模型 III.....	12
5.3.4 熵权 TOPSIS 组合模型评价三种方案.....	13
5.3.5 问题二的解答.....	17
5.4 问题三的模型.....	17
5.4.1 遗传算法.....	17
5.4.2 遗传算法的基本流程.....	18
5.4.3 模型建立与求解.....	20
六、结果分析.....	21
七、模型评价与推广.....	21
7.1 模型的优点.....	21
7.2 模型的缺点.....	22
7.3 模型的改进.....	22
八、附录.....	23
九、参考文献.....	29

1 一、问题重述

1.1 问题背景

当今俄罗斯与乌克兰的冲突已经进入到白日化阶段,通常情况下士兵上战场不会带上全部的物资,需要后方补给站进行物资补给,进行高效的物资补给策略将直接影响战争双方的主动权,因此战争中物资的补给方案的合理与否显得尤为关键。

1.2 已知条件

在一个补给站周围有四个部队需要进行物资补给,该补给站现有 6 架 A 类飞机, 10 架 B 类飞机,其中 A 类飞机飞行速度为 260km/h,最大承重为 13t,装卸货物所需时间均为 20min, B 类飞机飞行速度为 50km/h,最大承重为 20t,装卸货物需要 30min,卸货物需要 40min。一架飞机每次只能到一个部队运送货物且运送完成后需回到补给点装载货物后才能进行下一次运送任务。要求每个部队所可以接受的物资不得多与所需物资的 150%,且运送时间不能长于所要求的最短运送时间。

1.3 解决问题

问题一:由于旧补给站被炸毁,需选取新的补给站,在不考虑其他因素的条件下,只考虑如何最大程度给各个部队提供物资支持,建立合适的数学模型选取新补给点的地址。

问题二:选定的新补给站的位置是(10,30),建立数学模型确定飞机的调用方案。

问题三:各部队在接收到物资后便可前往下一个地点执行任务,要求部队不能携带超过所需 110%的物资,且在执行任务的过程中, A 类飞机有 5%-10%的损耗, B 类飞机有 3%-7%的损耗,建立数学模型确定转运方案。

2 二、问题分析

2.1 问题一分析

针对问题一，根据四个部队(A102,A122,B121,B404)所在的位置，不考虑其他的因素，即考虑平面坐标中四个部队与新补给站的距离，从而选取最合适的新补给站的位置。为了最大程度给各个部队提供支援，所以我们考虑建立最小圆覆盖模型，采用模拟退火算法，搜索出新补给站的最优位置。

2.2 问题二分析

针对问题二，题目要求以物资转运量最大为目标，以飞机调往四个部队的两种类型飞机数量为决策变量，以四个部队的物资需求量和最短需求时间为限制，通过改变约束条件，我们考虑建立多种线性整数规划模型。接着我们利用熵权TOPSIS 组合模型对多种线性整数规划模型进行评分，最终采用一种得分最高的整数规划模型，以期确定最优的飞机调用方案。

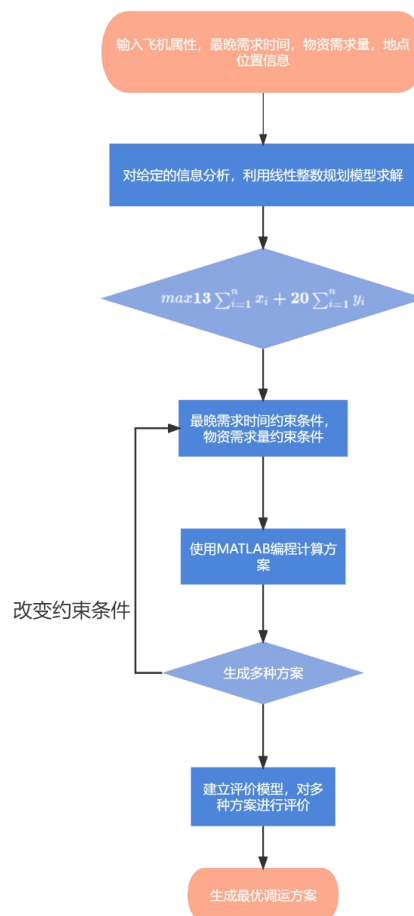


图 2-1 解决问题二的流程图

2.3 问题三分析

针对问题三，相比前两问，题目添加了部队不能携带超过所需 110% 的物资，且飞机在转运过程中存在损耗的约束条件，我们沿用第二问选择的最优整数规划模型，为了进一步提高数据的准确度和合理性，我们利用遗传算法对模型进行优化，进而确定转运方案。

3 三、模型假设

- 1、忽略飞机转运过程中天气的影响。
- 2、假设飞机到达目的地完成卸货任务后立即返回补给点。
- 3、假设第二问飞机转运过程中的不存在损耗问题。
- 4、假设飞机飞行距离即出发地与目的地的直线距离。
- 5、假设飞机在进行转运任务过程中，部队和补给点的位置均不发生改变。
- 6、忽略数据之间微小差距的影响，比如采用四舍六入五成双、取整等处理。

4 四、定义与符号说明

符号	定义
n	$n = 4$, 部队的个数
x_i	A 类飞机飞往各部队的架数
y_j	B 类飞机飞往各部队的架数
$lossA$	A 类飞机损耗 5%-10%
$lossB$	B 类飞机损耗 3%-7%

5 五、模型的创立与求解

5.1 准备工作

5.1.1 数据可视化

四个部队的平面坐标、物资需求量、最晚需求时间如图 5-1 所示。

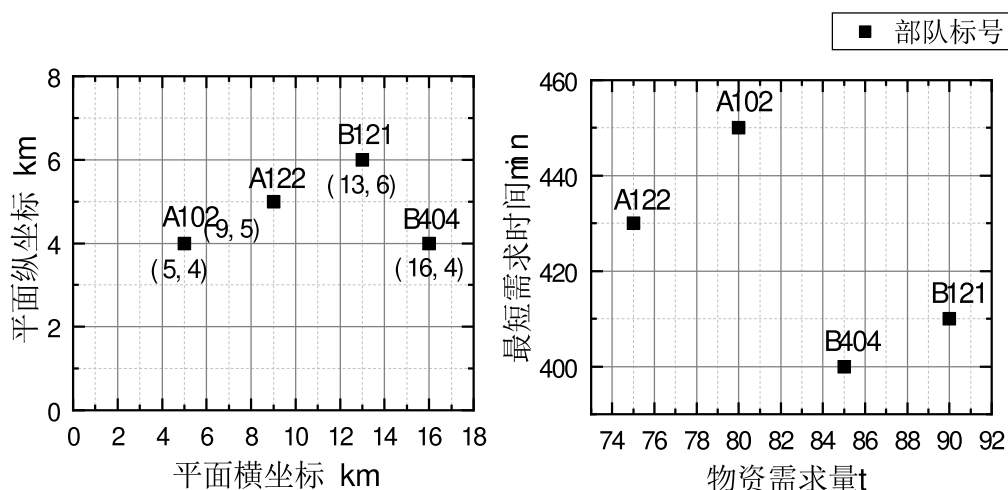


图 5-1 四个部队属性

A 类飞机和 B 类飞机的数量、载重、飞行速度、装卸时间如图 5-2 所示。

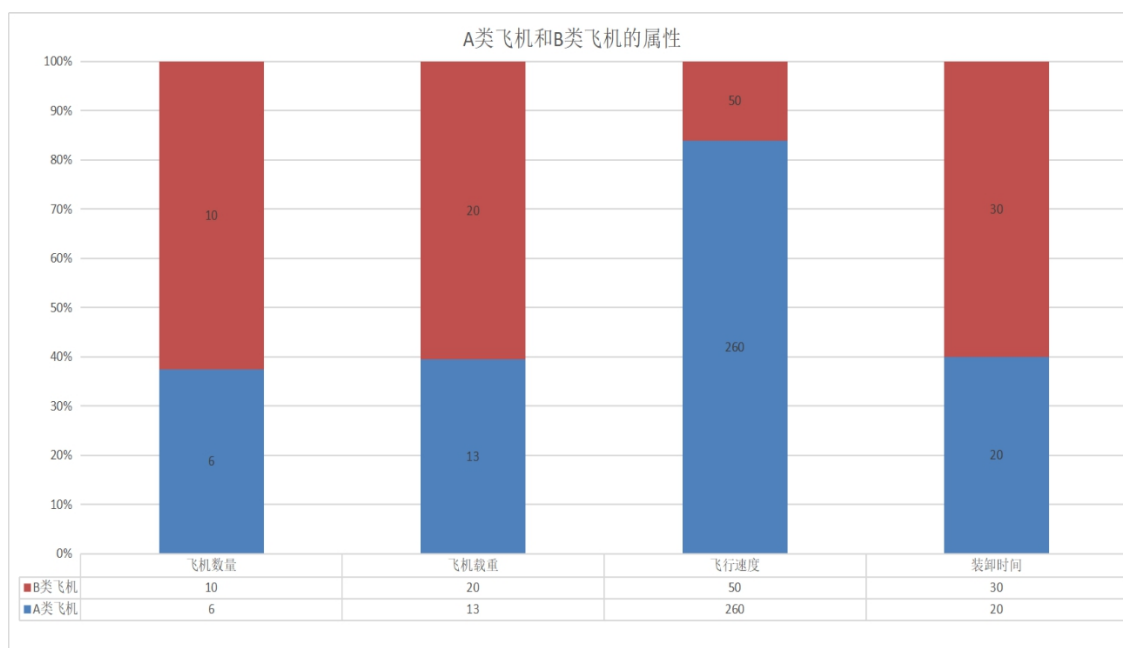


图 5-2 飞机属性

5.1.2 数据处理

通过编写 C++ 程序，计算得到补给站(10,30)和四个部队之间的平均距离为 25.5916 千米，补给站到最近的部队的距离为 24.1868 千米，到最远的部队的距离为 26.6833 千米，所以把补给站到最近点和最远点距离的差值 2.4965 千米记为 Δx ，A 类飞机飞行 Δx 需耗时 0.57611 分钟，B 类飞机飞行 Δx 需耗时 2.9958 分钟，相较于各点最晚需求时间相比，数据极小，可忽略不计。所以将补给站与四个部队的距离的平均和视为补给站到四个部队的距离。

计算飞机的飞行时间:

A类飞机到部队的时间(分钟)= $25.5916 \div 260 \times 60 = 5.9058$

A类飞机从装货起飞到目的地卸货后返回所需时间(分钟) = $20 + 5.9058 + 20 + 5.9058 = 51.8116$

B类飞机到部队的时间(分钟)= $25.5916 \div 50 \times 60 = 30.7099$

B类飞机从装货起飞到目的地卸货后返回所需时间(分钟) = $30 + 30.7099 + 40 + 30.7099 = 131.4198$

为了方便计算,同时,所求得的时间数据与部队最晚需求时间差距较大,所以我们对数据进行四舍五入取整。

声明:当最后一个或多个部队物资需求被满足时,此时飞机回航的时间不在计入总的转运时间内。

5.2 问题一的模型

5.2.1 最小圆覆盖算法

最小圆覆盖模型是常用的数学模型,它求出满足坐标系中某些特定点都在这个最小的圆内,即这些点连接所形成的封闭图形的最小外切圆。该最小外切圆半径 $\min(r)$ 为

$$r = cov(\sum_{i=0}^n p_i) \quad (1)$$

最小圆覆盖问题指平面上有 n 个点,给定 n 个点的坐标,找到一个半径最小的圆,将 n 个点全部包围,点可以在圆上。

5.2.2 模拟退火算法

我们所采用的模拟退火算法是一种元启发式技术^[1],其特征是能够在较短的时间内获得规划问题的满意解。该算法常用的参数包括初始温度(T_0),最终温度(t),冷却速率(r),每温度下的交互次数($nrep$)以及初始解数量(N)等。根据模拟退火算法原理,提高目标函数值的解总会被接受,而不能提高目标函数值得解则会按一定的概率(P)被接受。值取决于当前温度 和目标函数值。模拟退火算法这一独特的策略不仅能够使得算法的搜索过程跳出局域最优解,还能促使算法探索更广泛的解空间,进而增加该算法获得最优解的概率。标准版本的模拟退火算法可表示为如下过程:

- 1、初始化:取初始值 T_0 足够大,令 $T=T_0$,任取初始解 S_1 。
- 2、对当前温度 T ,重复第(3)~(6)步。
- 3、对当前解 S_1 随机扰动产生一个新解 S_2 。
- 4、计算 S_2 的增量 $df=f(S_2)-f(S_1)$,其中 $f(S_1)$ 为 S_1 的代价函数。
- 5、若 $df < 0$,则接受 S_2 作为新的当前解,即 $S_1=S_2$;否则计算 S_2 的接受概率 $\exp(-df/T)$,即随机产生(0,1)区间上均匀分布的随机数 $rand$,若 $\exp(-df/T) > rand$,也接受 S_2 作为新的当前解 $S_1=S_2$,否则保留当前解 S_1 。
- 6、如果满足终止条件 $Stop$,则输出当前解 S_1 为最优解,结束程序,种植条件 $Stop$ 通常取为在连续若干个 Metropolis 链中新解 S_2 都没有被接受

时终止算法或者是设定结束温度;否则按衰减函数衰减 T 后返回第(2)步。
 在所有枚举点中找到补给点到四个所给点中最远点的距离, 需要保证该距离最小, 即求圆的半径。

5.2.3 求解

利用 C++ 编写最小圆覆盖算法和模拟退火算法, 输入四个部队的平面坐标, 运行即可得到符合条件的最小外切圆的圆心坐标和半径。这个最小外切圆的圆心坐标即为新补给站修建地点的平面坐标。程序的运行结果为:

请输入四个点的坐标, 点坐标的 x 与 y 轴数据之间空一格, 没输入完一个坐标点, 就按下一次回车键.

5 4
 9 5
 13 6
 16 4
 最小圆的圆心为 (10.500, 4.000), 半径为 5.500 (单位: 千米)。

求解圆的方程为: ($r = 5.5$)

$$(x - 10.5)^2 + (y - 4.0)^2 = r^2 \quad (2)$$

综上所述, 我们新补给站的修建地点为 (10.5, 4.0)。

5.3 问题二的模型

为了选取最优的飞机调用方案, 我们通过增减或改变约束条件, 建立三种寻求飞机最优调用方案的模型。在数据处理中发现, 实际运输时间远小于最晚需求时间, 几乎不可能迟于最晚需求时间到达, 因此我们在建立规划模型时, 不把时间约束考虑在内。

5.3.1 模型 I

第一种方案, 我们选择在飞机第一次飞行时就满足一部分部队的需求, 即约束条件为各个部队接受的物资不大于所需物资的 150%。建立整数规划模型。

第一步, 确定最大运输量。

目标函数: 运输量最大, 即

$$\max 13 \sum_{i=1}^n x_i + 20 \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \quad (3)$$

约束条件如下:

(1) 第一次飞机起飞, A 类飞机一共有 6 架, B 类飞机一共有 10 架, 即

$$\sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n)$$

(2) 各部队的接收物资量都不超过自身物资需求量的 150%。

$$\begin{cases} 13x_1 + 20y_1 \leq 80 \times 1.5 \\ 13x_2 + 20y_2 \leq 75 \times 1.5 \\ 13x_3 + 20y_3 \leq 90 \times 1.5 \\ 13x_4 + 20y_4 \leq 85 \times 1.5 \end{cases} \quad (4)$$

(3) A 类飞机与 B 类飞机的个数为大于或者等于零的整数。

$$x_i, y_j \in N, (N \text{ 为非负整数})$$

决策变量： x_i, y_j 。

以数学语言描述如下：

$$\begin{aligned} & \max 13 \sum_{i=1}^n x_i + 20 \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \\ & s.t. \begin{cases} \sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n) \\ 13x_1 + 20y_1 \leq 80 \times 1.5 \\ 13x_2 + 20y_2 \leq 75 \times 1.5 \\ 13x_3 + 20y_3 \leq 90 \times 1.5 \\ 13x_4 + 20y_4 \leq 85 \times 1.5 \\ x_i, y_j \in N, (N \text{ 为非负整数}) \end{cases} \end{aligned} \quad (5)$$

将上述模型编辑成 MATLAB 程序，经过计算可得：

表 5-1 方案一第一次起飞

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	0	0	0	80	80
A122	2	0	26	49	75
B121	4	4	132	0	90
B404	0	6	120	0	85
总计	6	10	278	129	330

第二次运输时 B 类飞机尚未回航，只有 A 类飞机可进行转运任务。进行整数规划。

第一步，确定最大运输量。

目标函数：运输量最大，即

$$\max 13 \sum_{i=1}^{n-2} x_i, i \in (1, 2, \dots, n-2) \quad (6)$$

约束条件如下：

(1) 第二次飞机起飞，一共有 A 类飞机 6 架，即

$$\sum_{i=1}^2 x_i \leq 6, i \in (1, 2, \dots, 2) \quad (7)$$

(2) 各部队的接收物资量都不超过自身物资需求量的150%。

$$\begin{cases} 13x_1 \leq 80 \times 1.5 \\ 13x_2 \leq 75 \times 1.5 - 26 \end{cases} \quad (8)$$

(3) A 类飞机与 B 类飞机的个数为大于或者等于零的整数。

$$x_i, y_j \in N, (N \text{ 为非负整数})$$

决策变量： x_i, y_j 。

以数学语言描述如下：

$$\begin{aligned} & \max 13 \sum_{i=1}^{N-2} x_i, i \in (1, 2, \dots, N-2) \\ & s.t. \begin{cases} \sum_{i=1}^N x_i \leq 6, j \in (1, 2, \dots, N-2) \\ 13x_1 \leq 80 \times 1.5 \\ 13x_2 \leq 75 \times 1.5 - 26 \\ x_i, y_j \in N, (N \text{ 为非负整数}) \end{cases} \end{aligned} \quad (9)$$

将上述模型编辑成 MATLAB 程序，经过计算可得：

表 5-2 方案一第二次起飞

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	0	0	0	80	80
A122	8	0	104	0	75
B121	4	4	132	0	90
B404	0	6	120	0	85
总计	12	10	356	80	330

此时只有 A101 还需物资 80t，且 B 类飞机仍未返回补给站，所以我们再次使用 A 类飞机进行转运，但是 6 架 A 类飞机最多只能转运 78t 物资，仍未达到 A101 物资需求量，还需要进行第四次转运，尽管此时 B 类飞机已经返回，但是通过计算使用 A 类飞机转运，耗时更少。

综上所述，这种方法在 201 分钟内共转运 447t 物资。

5.3.2 模型 II

很明显，第一种方案是存在一定迟钝性的，所以我们在第一种方案的基础上加以改良。

第二种方案，由观察可知，各地所需物资均不超过 100t 换言之，5 架 B 类飞机便可以使得一个部队需求物资被满足，所以增加 B 类飞机飞向一个部队的数量不超过五的约束。即：

$$y_j \leq 5, j \in (1, 2, \dots, n)$$

决策变量： x_i, y_j 。

以数学语言描述如下：

$$\begin{aligned} & \max 13 \sum_{i=1}^n x_i + 20 \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \\ & s.t. \begin{cases} \sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n) \\ 13x_1 + 20y_1 \leq 80 \times 1.5 \\ 13x_2 + 20y_2 \leq 75 \times 1.5 \\ 13x_3 + 20y_3 \leq 90 \times 1.5 \\ 13x_4 + 20y_4 \leq 85 \times 1.5 \\ y_j \leq 5, j \in (1, 2, \dots, n) \\ x_i, y_j \in N, (N \text{ 为非负整数}) \end{cases} \end{aligned} \quad (10)$$

将上述模型编辑成 MATLAB 程序，经过计算可得：

表 5-3 方案二的第一次起飞

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	0	0	0	80	80
A122	4	0	52	23	75
B121	2	5	126	0	90
B404	0	5	100	0	85
总计	6	10	278	103	330

此时部队 B121 和部队 B404 已满足物资需求量，所以在接下来的转运中，我们只考虑部队 A101 和部队 A122。

A122 部队明显只需要两架 A 类飞机就可以了，所以限定到达 2 部队的飞机不大于 2，进行整数规划。

第一步，确定最大运输量。

目标函数：运输量最大，即

$$\max 13 \sum_{i=1}^{n-2} x_i, i \in (1, 2, \dots, n-2) \quad (11)$$

约束条件如下：

(1) 第二次飞机起飞，一共有 A 类飞机 6 架，即

$$\sum_{i=1}^2 x_i \leq 6, i \in (1, 2, \dots, 2) \quad (12)$$

(2) 各部队的接收物资量都不超过自身物资需求量的 150%。

$$\begin{cases} 13x_1 \leq 80 \times 1.5 \\ 13x_2 \leq 75 \times 1.5 - 42 \end{cases} \quad (13)$$

(3) A 类飞机与 B 类飞机的个数为大于或者等于零的整数。

$$x_i, \in N, (N \text{ 为非负整数})$$

(4) 限定不大于 2

$$x_2 \leq 2$$

决策变量: x_i, y_j 。

以数学语言描述如下:

$$\begin{aligned} \max & 13 \sum_{i=1}^{N-2} x_i, i \in (1, 2, \dots, N-2) \\ \text{s.t.} & \begin{cases} \sum_{i=1}^N x_i \leq 6, i \in (1, 2, \dots, N-2) \\ 13x_1 \leq 80 \times 1.5 \\ 13x_2 \leq 75 \times 1.5 - 42 \\ x_2 \leq 2 \\ x_i, y_j \in N, (N \text{ 为非负整数}) \end{cases} \end{aligned} \quad (14)$$

将上述模型编辑成 MATLAB 程序, 计算结果如表 5-4 所示。

表 5-4 方案二的第二次起飞

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	4	0	52	28	80
A122	6	0	78	0	75
B121	2	5	126	0	90
B404	0	5	100	0	85
总计	12	10	356	28	330

此时, 部队 A101 还缺物资 28t, 再派 A 类飞机三架就可以满足所有部队的需求, 总耗时 150min, 总运量 395t。

5.3.3 模型 III

第三种方案, 我们可以看到在满足第一次就使得部分部队需求被满足不一定是最好的方案, 所以我们改变方案, 采用第一次飞时尽量让每个部队都得到物资的决策, 进行整数规划。

根据战场实际情况和常识经验, 期望实际转运物资与需求量尽可能接近, 满足各部队的基本需求, 提高转运效率, 优化分配方案。我们在多次实验之后发现约束条件为各部队接收物资不超过自身物资需求量的 110% 时, 转运效率较高。

第一步, 确定最大运输量。

目标函数: 运输量最大, 即

$$\max 13 \sum_{i=1}^n x_i + 20 \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \quad (15)$$

约束条件如下:

(1) 第一次飞机起飞, A 类飞机一共有 6 架, B 类飞机一共有 10 架, 即

$$\sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n)$$

(2) 各部队的接收物资量都不超过自身物资需求量的 150%。

$$\begin{cases} 13x_1 + 20y_1 \leq 80 \times 1.1 \\ 13x_2 + 20y_2 \leq 75 \times 1.1 \\ 13x_3 + 20y_3 \leq 90 \times 1.1 \\ 13x_4 + 20y_4 \leq 85 \times 1.1 \end{cases} \quad (16)$$

(3) A 类飞机与 B 类飞机的个数为大于或者等于零的整数。

$$x_i, y_j \in N, (N \text{ 为非负整数})$$

决策变量: x_i, y_j 。

以数学语言描述如下:

$$\begin{aligned} & \max 13 \sum_{i=1}^n x_i + 20 \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \\ & s.t. \begin{cases} \sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n) \\ 13x_1 + 20y_1 \leq 80 \times 1.1 \\ 13x_2 + 20y_2 \leq 75 \times 1.1 \\ 13x_3 + 20y_3 \leq 90 \times 1.1 \\ 13x_4 + 20y_4 \leq 85 \times 1.1 \\ x_i, y_j \in N, (N \text{ 为非负整数}) \end{cases} \end{aligned} \quad (17)$$

将上述模型编辑成 MATLAB 程序, 经过计算可得:

表 5-5 方案三的第一次起飞

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	1	0	13	67	80
A122	3	2	79	0	75
B121	1	4	93	0	90
B404	1	4	93	0	85
总计	6	10	278	67	330

此时, 我们发现只有部队 A101 还差 67t 达到物资需求量, 且 B 类飞机还未返回, 于是我们再次调用 6 架 A 类飞机, 可以满足全部部队的物资需求。耗时 101 分钟, 共转运物资 356t。

5.3.4 熵权 TOPSIS 组合模型评价三种方案

三种方案的耗时和运输量如下表 5-6 所示, 可以看出, 三种方案的耗时均小于各个部队的最晚需求时间。

表 5-6 三个方案的耗时、运量

	耗时（分钟）	运量（吨）
方案一	201	447
方案二	150	395
方案三	101	356

TOPSIS 法是 C.L.Hwang 和 K.Yoon 在 1981 年首次提出的模型，该模型首次引入了正负理想解的概念，充分解释正理想解和负理想解间存在的关系以及所适用的场合。正理想解理论上是决策的最优方案，在功能和效果上都取到最大值。负理想解与之相反，是决策中最差的方案，这使得方案的可行性最小或风险最大。把正理想方案和负理想方案之间的取值作为一个区间，在所有备选方案中选择最优解，选择的方案必须远离负理想方案，同时和正理想方案距离最相近，这样就是备选方案中的最优决策。TOPSIS 模型在评价方案的优劣性时引用欧几里距离进行测算，欧几里距离就是与正理想解和负理想解之间的距离。尽管 TOPSIS 方法分别计算矩阵数据的最优解和最差解是复杂且繁琐的，但其通过对风险权重的确定，更加有利于对项目投资的决策，同时该方法的优势也在于对于评价对象和数据指标的限制也要求较少，能够广泛适用。熵权 TOPSIS 决策模型^[2]一般是先由熵权法来确定评价指标的权重，之后用 TOPSIS 法对评价对象进行排序。

5.3.4.1 熵权法

(1) 首先对数据进行归一化处理：

由于各项指标计量单位并不统一，因此在计算综合权重前先要对它们进行标准化处理，即把指标的绝对值转化为相对值，并令 $x_{ij} = |x_{ij}|$ ，从而解决各项不同质指标值的同质化问题。而且，由于正向指标和负向指标数值代表的含义不同（正向指标数值越高越好，负向指标数值越低越好），因此，对于高低指标我们用不同的算法进行数据标准化处理。其具体方法如下：

对于正向指标：

$$x_{ij} = 0.998 \frac{x_{ij} - \min\{x_{1j}, \dots, x_{nj}\}}{\max\{x_{1j}, \dots, x_{nj}\} - \min\{x_{1j}, \dots, x_{nj}\}} + 0.002 \quad (18)$$

对于负向指标：

$$x_{ij} = 0.998 \frac{\max\{x_{1j}, \dots, x_{nj}\} - x_{ij}}{\max\{x_{1j}, \dots, x_{nj}\} - \min\{x_{1j}, \dots, x_{nj}\}} + 0.002 \quad (19)$$

(2) 计算第 j 项指标下第 i 方案指标值的比重 p_{ij} ：

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}} (j=1, 2, \dots, m) \quad (20)$$

(3) 计算第 j 项指标的熵值 e_j ：

$$e_j = -k \sum_{i=1}^n p_{ij} \ln p_{ij} \quad (21)$$

其中 $k = 1 / \ln(n)$ ，满足 $e_j \geq 0$

(4) 计算信息熵冗余度：

$$g_j = 1 - e_j \quad (22)$$

(5) 计算各项指标权重

$$w_j = \frac{g_j}{\sum_{j=1}^m g_j} \quad (23)$$

(6) 计算综合得分：

$$s_i = \sum_{j=1}^m w_j x_{ij} \quad (24)$$

问题二中耗时为负向指标，运量越贴近需求量越好，应为点指标，但三种方案的运量均大于需求量，所以按负指标来计算权重。将上述模型编辑成 MATLAB 程序，经过计算可得图 5-3：

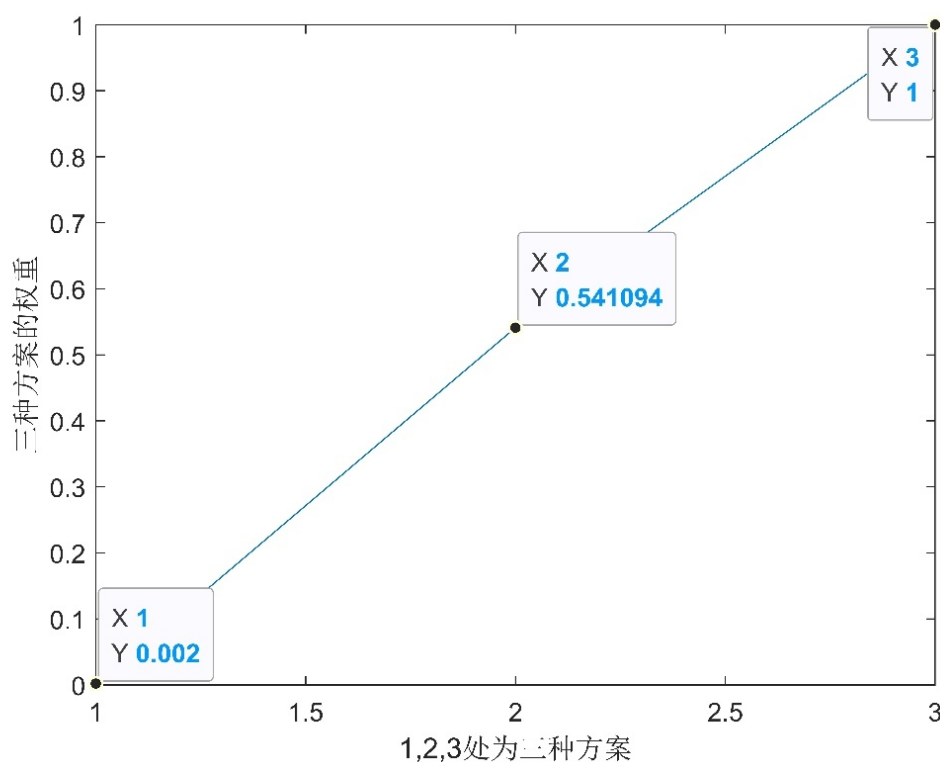


图 5-3 三种方案求得三种方案的权重

5.3.4.2 TOPSIS 法

1、TOPSIS 法简介

TOPSIS 算法是一种常见的综合评价方法，其能充分利用原始数据的信息，其结果能精确地反应各评价方案之间的差距。逼近理想解排序法 (the technique for order preference by similarity to an ideal solution, TOPSIS) 是一种通过比较样本值与理想值的距离实现综合评价的方法。

逼近理想解排序法采用相对接近度来表征各个评价对象与参考点的距离。首先在空间确定出参考点，包括最优和最差点，然后计算各个评价对象与参考点的距离，与最优点越近或与最差点越远说明被评价对象的综合特性越好。

2、TOPSIS 法基本思路

2.1 统一指标类型

评价指标可以分为正向指标和负向指标，一般可都统一为正向指标。注意各种类型的指标正向化。

2.2 标准化处理

假设有 n 个要评价的对象， m 个已经正向化的评价指标，则可以构成一个 $m \times n$ 的矩阵 $(x_{ij})_{m \times n}$ ，采用下式进行指标标准化处理：

$$Z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}} \quad (25)$$

2.3 确定正理想解和负理想解

确定各指标的理想解，有正理想解。

$$Z^+ = (Z_1^+, Z_2^+, \dots, Z_n^+) = \max_{1 \leq i \leq n} Z_{ij}, j = 1, 2, \dots, m \quad (26)$$

负理想解。

$$Z^- = (Z_1^-, Z_2^-, \dots, Z_n^-) = \min_{1 \leq i \leq n} Z_{ij}, j = 1, 2, \dots, m \quad (27)$$

2.4 计算距离

计算各指标离正负理想解的距离，有正理想解距离。

$$D_i^+ = \sqrt{\sum_{j=1}^m (Z_j^+ - Z_{ij})^2} \quad (28)$$

负理想解距离。

$$D_i^- = \sqrt{\sum_{j=1}^m (Z_j^- - Z_{ij})^2} \quad (29)$$

2.5 计算相对接近度

计算各指标的相对接近度，即

$$S_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (30)$$

采用熵权法得出三种方案的权重，将上述模型编辑成 MATLAB 程序，经过计算可得：

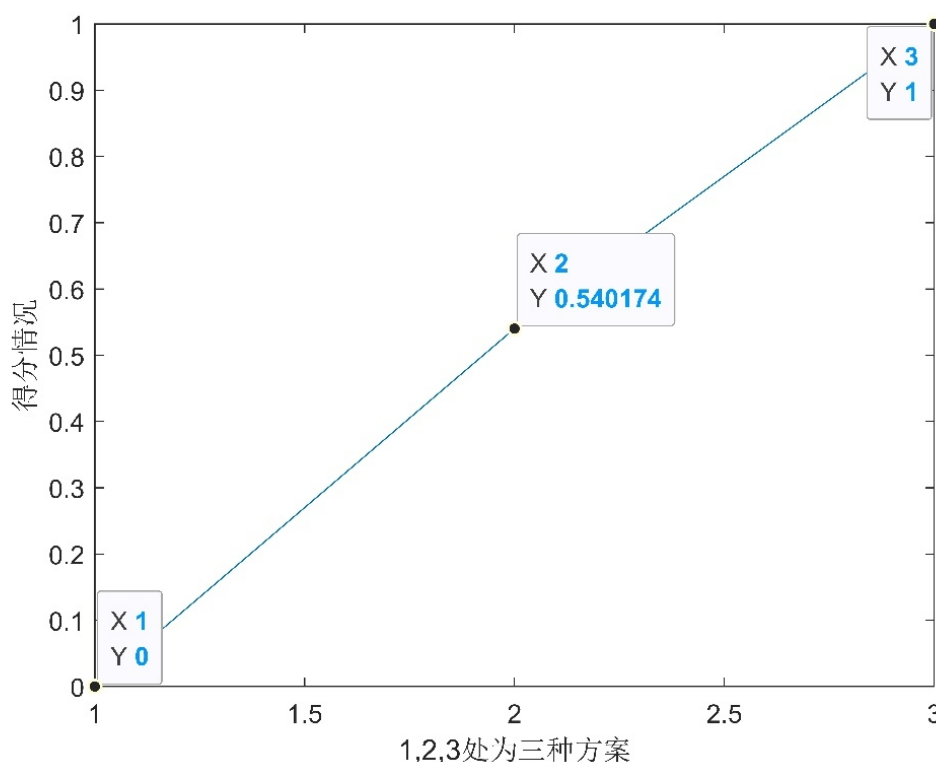


图 5-4 三种方案的得分情况

三种方案的得分为：第一种方案的得分为 0；第二种方案的得分为 0.540174；第三种方案的得分为 1。第三种方案的得分最高，所以选择第三种方案。

5.3.5 问题二的解答

最后我们选择的飞机调用方案是：首先，飞往 A101 部队，A 类飞机 1 架，B 类飞机 0 架；飞往 A122 部队，A 类飞机 3 架，B 类飞机 2 架；飞往 B121 部队，A 类飞机 1 架，B 类飞机 4 架；飞往 B404 部队，A 类飞机 1 架，B 类飞机 4 架。然后，当 A 类飞机返回后，立即进行第二次转运，将 6 架 A 类飞机全部飞往 A101 部队。总耗时 101 分钟，总运量 356 吨。

5.4 问题三模型

5.4.1 遗传算法

遗传算法^[3] (Genetic Algorithm, GA) 是进化计算的一部分，是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。该算法简单、通用，鲁棒性强，适于并行处理。它必须做以下操作：初始群体的产生、求每一个体的适应度、根据适者生存的原则选择优良个体、被选出的优良个体两两配对，通过随机交叉其染色体的基因并随机变异

某些染色体的基因生成下一代群体,按此方法使群体逐代进化,直到满足进化终止条件。

遗传算法的特点

1. 以决策变量的编码作为运算对象。

传统的优化算法往往直接利用决策变量的实际值本身来进行优化计算,但遗传算法是使用决策变量的某种形式的编码作为运算对象。这种对决策变量的编码处理方式,使得我们在优化计算中可借鉴生物学中染色体和基因等概念,可以模仿自然界中生物的遗传和进化激励,也可以很方便地应用遗传操作算子。

2. 直接以适应度作为搜索信息。

传统的优化算法不仅需要利用目标函数值,而且搜索过程往往受目标函数的连续性约束,有可能还需要满足“目标函数的导数必须存在”的要求以确定搜索方向。

遗传算法仅使用由目标函数值变换来的适应度函数值就可确定进一步的搜索范围,无需目标函数的导数值等其他辅助信息。直接利用目标函数值或个体适应度值也可以将搜索范围集中到适应度较高部分的搜索空间中,从而提高搜索效率。

3. 使用多个点的搜索信息,具有隐含并行性。

传统的优化算法往往是从解空间的一个初始点开始最优解的迭代搜索过程。单个点所提供的搜索信息不多,所以搜索效率不高,还有可能陷入局部最优解而停滞;

遗传算法从由很多个体组成的初始种群开始最优解的搜索过程,而不是从单个个体开始搜索。对初始群体进行的、选择、交叉、变异等运算,产生出新一代群体,其中包括了许多群体信息。这些信息可以避免搜索一些不必要的点,从而避免陷入局部最优,逐步逼近全局最优解。

4. 使用概率搜索而非确定性规则。

传统的优化算法往往使用确定性的搜索方法,一个搜索点到另一个搜索点的转移有确定的转移方向和转移关系,这种确定性可能使得搜索达不到最优店,限制了算法的应用范围。

遗传算法是一种自适应搜索技术,其选择、交叉、变异等运算都是以一种概率方式进行的,增加了搜索过程的灵活性,而且能以较大概率收敛于最优解,具有较好的全局优化求解能力。但,交叉概率、变异概率等参数也会影响算法的搜索结果和搜索效率,所以如何选择遗传算法的参数在其应用中是一个比较重要的问题。

综上,由于遗传算法的整体搜索策略和优化搜索方式在计算时不依赖于梯度信息或其他辅助知识,只要求解影响搜索方向的目标函数和相应的适应度函数,所以遗传算法提供了一种求解复杂系统问题的通用框架。它不依赖于问题的具体领域,对问题的种类有很强的鲁棒性,所以广泛应用于各种领域

5.4.2 遗传算法的基本流程

1.通过随机方式产生若干由确定长度(长度与待求解问题的精度有关)编码的初始群体;

2.通过适应度函数对每个个体进行评价,选择适应度值高的个体参与遗传操作,适应度低的个体被淘汰;

3.经遗传操作（复制、交叉、变异）的个体集合形成新一代种群，直到满足停止准则（进化代数 $GEN \geq ?$ ）；

4. 将后代中表现最好的个体作为遗传算法的执行结果。

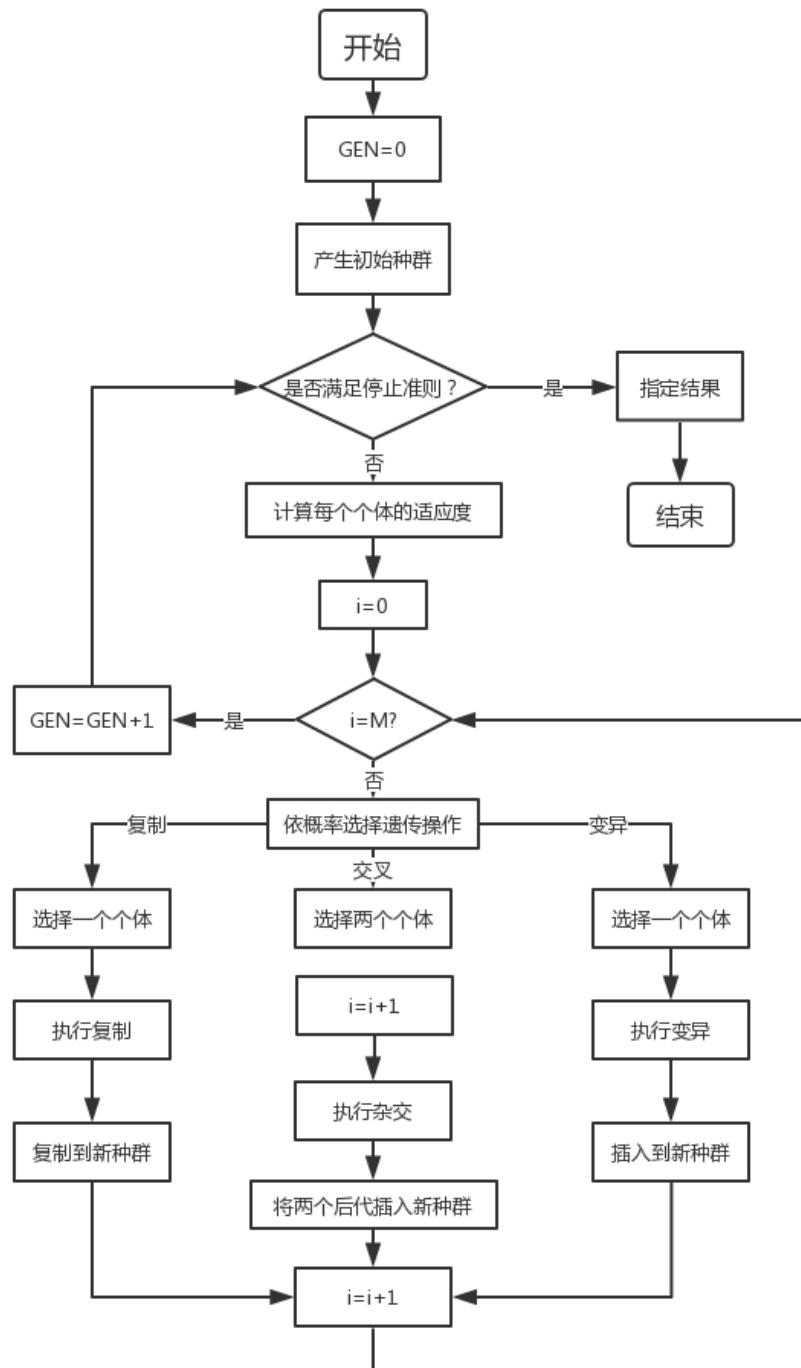


图 5-5 遗传算法的流程图

5.4.3 模型建立与求解

我们沿用了问题二中选出的整数规划模型，并用遗传算法进行优化，遗传算法的参数设置为：初始种群个数 50；最大迭代次数 500；变异概率 0.01；交叉概率 0.5。整数规划模型的数学表达式为：

$$\begin{aligned} & \max(1 - \text{loss}A) \sum_{i=1}^n x_i + (1 - \text{loss}B) \sum_{j=1}^n y_j, i, j \in (1, 2, \dots, n) \\ & \left\{ \begin{array}{l} \sum_{i=1}^n x_i \leq 6, \sum_{j=1}^n y_j \leq 10, j \in (1, 2, \dots, n) \\ (1 - \text{loss}A)x_1 + (1 - \text{loss}B)y_1 \leq 80 \times 1.1 \\ (1 - \text{loss}A)x_2 + (1 - \text{loss}B)y_2 \leq 75 \times 1.1 \\ (1 - \text{loss}A)x_3 + (1 - \text{loss}B)y_3 \leq 90 \times 1.1 \\ s.t. \left\{ \begin{array}{l} (1 - \text{loss}A)x_4 + (1 - \text{loss}B)y_4 \leq 85 \times 1.1 \\ y_j \leq 4 \\ x_i, y_j \in N, (N \text{ 为非负整数}) \\ 0.05 \leq \text{loss}A \leq 0.10 \\ 0.03 \leq \text{loss}B \leq 0.07 \end{array} \right. \end{array} \right. \quad (31) \end{aligned}$$

由于损耗率的不可确定性，我们分别取损耗率最大和损耗率最小进行规划，由飞机调度的决策者根据可承担风险的大小进行选择。我们是利用数据分析软件 SPSSPRO 进行计算结果的。

风险最小，运输量损失最大：

表 5-7 最大损耗率下的飞机调动方案

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	2	2	63.5	16.5	80
A122	2	1	44.1	30.9	75
B121	1	4	89.95	0.05	90
B404	1	3	70.55	14.45	85
总计	6	10	268.1	61.9	330

此时，由于 B121 部队未运输量 0.05 吨相比于物资需求量过小，不再向它运送物资。返回的 6 架 A 类飞机，再派往 A101 部队 2 架 A 类飞机，派往 A122 部队 3 架 A 类飞机，派往 B404 部队 1 架飞机。最后，在派往 B404 部队 1 架飞机，即可完成运输任务。此时耗时为 150 分钟，总运输量为 369 吨。

风险最小，运输量损失最大：

表 5-8 最小功率下的飞机调动方案

	A 类飞机数量	B 类飞机数量	已运输量	未运输量	物资需求量
A101	0	3	55.8	24.2	80
A122	0	4	74.4	0.6	75
B121	3	2	72.3	17.7	90
B404	3	1	53.7	31.3	85
总计	6	10	256.2	73.8	330

此时，A122 部队还需要 0.6 吨物资，但相比于其需要的总物资可忽略不计，所以我们不再向 A122 部队运送物资。在 A 类飞机返回后，派 2 架去 A101 部队，2 架去 B121 部队，2 架去 B404 部队，在问题二中已经算出，第三次还是需要派 A 类飞机，所以最后再派一架 A 类飞机去 B404 部队。在此类方法中耗时 150 分钟，运量为 356 吨。

6 六、结果分析

问题一中采用最小圆覆盖这种算法来求解补给站的最佳位置，具有较好的拟和性，同时采用模拟退火算法来优化模型，使得最终的结果更为准确合理，但从主观的角度来看，预选取的补给站位置与其他部队的位置相距较近且相距距离接近，是较为合理的结果。

问题二中我们映射了贪心算法的理念，但又不完全照搬，循序渐进的提出了三种模型，并采用熵权 TOPSIS 组合模型来对三种模型进行评估，结果也如我们所料，经过改进的第三种模型在耗时和运量上更具优势，转运效率更高。

问题二和问题三中的物资得到 356 吨，最低需求为 330 吨，超出了最低需求的 79%，在满足需求的同时，尽量减少了物资运输量，是相对比较合理的调用方案。问题二耗时 101 分钟，为最早需求时间的 25%；问题三耗时 150 分钟，是最早需求时间的 37.5%，都以较短的时间完成了转运任务。

综上所述，我们的模型是较为合理的模型，结果的准确性较高。

7 七、模型评价与推广

7.1 模型的优点

问题一中的模型采用了模拟退火算法，可以突破爬山算法的局限性，获得全局最优解，且初始值与最终解都是随机选取的，它们之间毫无关联，因此具备较

好的鲁棒性, 抗外界干扰能力强。同时, 我们增加了迭代次数, 提高最优解的准确性。

问题二中采用了整数规划, 并且提出了多种方案, 利用熵权 TOPSIS 法进行评价, 有助于选取更优的飞机调用方案。

问题三中利用遗传算法求解, 具备较好的鲁棒性, 且适合求解复杂的优化问题。

7.2 模型的缺点

问题一中的模型采用了模拟退火算法, 算法受冷却速率的影响较大, 冷却速率慢, 可以获得更优的解, 但也会花费大量的时间; 冷却速度快, 可能得到最优解, 但也有可能跳过最优解。

问题二中的整数规划模型在计算后忽略了时间的约束条件, 使得整体的模型看起来没有那么完整, 还需要借助外界计算来完善模型。

问题三中的模型采用遗传算法, 收敛速度较慢, 且局部搜索能力较差, 无确定的终止准则。

7.3 模型的改进

针对问题一中的模型, 我们可以采用规划模型来优化搜索新补给站位置的模型。针对问题二中的模型, 我们可以使用优化算法 (例如粒子群算法) 来优化规划模型。问题三中的模型, 我们可以采用模糊规划的方式来选取飞机的调用方案, 以此应对飞机损耗率的不确定性。

8 八、附录

1、C++数据处理。

```

1. #include<iostream>
2. #include<math.h>
3. using namespace std;
4. double s,smin,v1,v2,t1,t2,tA,tB;
5. double distance(double x, double y)
6. {
7. s = sqrt((10-x)*(10-x)+(30-y)*(30-y));
8. return s;
9. }
10.int main()
11.{
12.double x1=5,x2=4,x3=9,x4=5,x5=13,x6=6,x7=16,x8=4;
13.cout << "据 A102 距离:" << distance(x1,x2) <<endl;
14.cout << "据 A122 距离:"<< distance(x3,x4) <<endl;
15.cout << "据 B121 距离:" << distance(x5,x6) <<endl;
16.cout << "据 B404 距离:" <<distance(x7,x8) <<endl;
17.cout << "平均值
   为:" << (distance(x1,x2)+distance(x3,x4)+distance(x5,x6)+distance(x7,
   x8))/4 << endl;
18.smin = distance (x3,x4)-distance(x6,x7);
19.t1=smin/260*60;
20.t2=smin/50*60;
21.tA=(distance(x1,x2)+distance(x3,x4)+distance(x5,x6)+distance(x7,x8))
   /4/260*60;
22.tB=(distance(x1,x2)+distance(x3,x4)+distance(x5,x6)+distance(x7,x8))
   /4/50*60;
23.cout <<"飞机 A 到最远点与最近点时间差:"<<t1<<endl;
24.cout << "飞机 B 到最远点与最近点时间差:"<<t2<<endl;
25.cout <<"飞机 A 到平均距离点所需时间:"<<tA<<endl;
26.cout <<"飞机 B 到平均距离点所需时间:"<<tB<<endl;
27.cout <<"飞机 A 和飞机 B 到最远点与最近点时间差<<飞机 A 到平均距离点所需时间
   " << endl;
28.return 0;
29.}

```

2、C++实现最小圆覆盖算法和模拟退火算法，求新补给站修建地点(在 Visual Studio 2019 环境下实现)。

```

1. #define _CRT_SECURE_NO_WARNINGS
2.

```

```

3.  #include<iostream>
4.  #include<cstring>
5.  #include<algorithm>
6.  #include<cstdio>
7.  #include<cmath>
8.  using namespace std;
9.  const double eps = 1e-8;
10. struct POINT {
11.     double x, y, z;
12. }p[510];
13. POINT op;//最小圆的圆心
14. int n;
15. inline double dist(POINT& a, POINT& b) {
16.     return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y -
b.y));
17. }
18. void solve() {
19.     double ret, delta = 100.0;
20.     double maxDis, tempDis;
21.     int i, id;
22.     while (delta > eps) {
23.         id = 0;
24.         maxDis = dist(op, p[id]);
25.         for (i = 1; i < n; i++) {
26.             tempDis = dist(op, p[i]);
27.             if (tempDis > maxDis) {
28.                 maxDis = tempDis;
29.                 id = i;
30.             }
31.         }
32.         ret = maxDis;
33.         op.x += (p[id].x - op.x) / maxDis * delta;
34.         op.y += (p[id].y - op.y) / maxDis * delta;
35.         delta *= 0.98;
36.     }
37.
38.     printf("最小圆的圆心为(%.3f, %.3f),半径为 %.3f (单位: 千米)。
\n", op.x, op.y, ret);
39. }
40. int main() {
41.     n = 4;
42.     op.x = op.y = 0;
43.     cout << "请输入四个点的坐标,点坐标的 x 与 y 轴数据之间空一格,没输入完
一个坐标点,就按下一次回车键." << endl;

```



```

44.     for (int i = 0; i < n; i++) {
45.         scanf("%lf%lf", &p[i].x, &p[i].y);
46.         op.x += p[i].x;
47.         op.y += p[i].y;
48.     }
49.     op.x /= n;
50.     op.y /= n;
51.     solve();
52.
53.     return 0;
54. }

```

3、方案一飞机第一次起飞的调用方案

```

1.     clc,clear
2.     f=[-13; -20; -13; -20;-13; -20;-13; -20];
3.     a=[0,1,0,1,0,1,0,1;1,0,1,0,1,0,1,0; 13,20,0,0,0,0,0,0;0,0,13,20,0,
0,0,0;0,0,0,0,13,20,0,0;0,0,0,0,0,0,13,20]; b=[10,6,80*1.5,75*1.5,90*1.
5,85*1.5];
4.     aeq=[1,1,1,1,1,1,1,1];
5.     beq=16;
6.     intcon = [1:8]
7.     [x,y]=intlinprog(f,intcon,a,b,aeq,beq,zeros(8,1));
8.     x, y=-y

```

4、方案一飞机第二次起飞的调用方案

```

1.     clc,clear
2.     f=[-13; -13];
3.     a=[1,1;13,0;0,13]; b=[6,80*1.5,75*1.5-26];
4.     aeq=[1,1];
5.     beq=6;
6.     intcon = [1:2]
7.     [x,y]=intlinprog(f,intcon,a,b,aeq,beq,zeros(2,1));
8.     x, y=-y

```

5、方案二飞机第一次起飞的调用方案

```

1.     clc,clear
2.     f=[-13; -20; -13; -20;-13; -20;-13; -20];
3.     a=[0,1,0,0,0,0,0,0;0,0,0,1,0,0,0,0;0,0,0,0,0,1,0,0;0,0,0,0,0,0,0,0,
1;0,1,0,1,0,1,0,1;1,0,1,0,1,0,1,0;13,20,0,0,0,0,0,0;0,0,13,20,0,0,0,0;0,
0,0,0,13,20,0,0;0,0,0,0,0,0,13,20]; b=[5,5,5,5,10,6,80*1.5,75*1.5,90*1.
5,85*1.5];
4.     aeq=[1,1,1,1,1,1,1,1];
5.     beq=16;
6.     intcon = [1:8]
7.     [x,y]=intlinprog(f,intcon,a,b,aeq,beq,zeros(8,1));
8.     x, y=-y

```

5、方案二飞机第二次起飞的调用方案

```

1.   clc,clear
2.   f=[-13; -13];
3.   a=[0,1;1,1;13,0;0,13]; b=[2,6,80*1.5,75*1.5-42];
4.   aeq=[1,1];
5.   beq=6;
6.   intcon = [1:2]
7.   [x,y]=intlinprog(f,intcon,a,b,aeq,beq,zeros(2,1));
8.   x, y=-y

```

6、方案三飞机第一次起飞的调用方案

```

1.   clc,clear
2.   f=[-13; -20; -13; -20;-13; -20;-13; -20];
3.   a=[0,1,0,1,0,1,0,1;1,0,1,0,1,0,1,0; 13,20,0,0,0,0,0,0;0,0,13,20,0,0,0,0;0,0,0,0,13,20,0,0;0,0,0,0,0,0,13,20]; b=[10,6,80*1.1,75*1.1,90*1.1,85*1.1];
4.   aeq=[1,1,1,1,1,1,1,1];
5.   beq=16;
6.   intcon = [1:8]
7.   [x,y]=intlinprog(f,intcon,a,b,aeq,beq,zeros(8,1));
8.   x, y=-y

```

7、熵权法代码

```

1.   clear all;
2.   clc;
3.   x = xlsread('data1.xlsx')%导入数据
4.   ind = ones(size(x,2),1)%正向写 1, 负向写 2
5.   ind(1) = 2;
6.   ind(2) = 2;
7.   [n,m]=size(x); % n 个样本, m 个指标 %%数据的归一化处理
8.   for i=1:m
9.       if ind(i)==1 %正向指标归一化
10.          X(:,i)=guiyi(x(:,i),1,0.002,1); %若归一化到[0,1], 0 会出问题
11.       else %负向指标归一化
12.          X(:,i)=guiyi(x(:,i),2,0.002,1);
13.       end
14.   end
15.   %%计算第 j 个指标下, 第 i 个样本占该指标的比重 p(i,j)
16.   for i=1:n;
17.       for j=1:m;
18.          p(i,j) = X(i,j)/sum(X(:,j));
19.       end
20.   end
21.   %%计算第 j 个指标的熵值 e(j)
22.   k=1/log(n);
23.   for j=1:m

```

```

24.         e(j)=-k*sum(p(:,j).*log(p(:,j)));
25.     end
26.     d=ones(1,m)-e; %计算信息熵冗余度
27.     w=d./sum(d); %求权值 w
28.     s=X*w'; %求综合得分
29.
30.     plot(1:3,s)
31.
32.     %以下为函数代码
33.     function y=guiyi(x,type,ymin,ymax)
34.     %实现正向或负向指标归一化, 返回归一化后的数据矩阵
35.     %x 为原始数据矩阵, 一行代表一个样本, 每列对应一个指标
36.     %type 设定正向指标 1, 负向指标 2
37.     %ymin,ymax 为归一化的区间端点
38.     [n,m]=size(x);
39.     y=zeros(n,m);
40.     xmin=min(x);
41.     xmax=max(x);
42.     switch type
43.         case 1
44.             for j=1:m
45.                 y(:,j)=(ymax-ymin)*(x(:,j)-xmin(j))/(xmax(j)-xmin(j))
+ymin;
46.             end
47.         case 2
48.             for j=1:m
49.                 y(:,j)=(ymax-ymin)*(xmax(j)-x(:,j))/(xmax(j)-xmin(j))
+ymin;
50.             end
51.     end

```

8、TOPSIS 代码

```

1.     %% 构建决策矩阵
2.     A = [0.002
3.         0.541094
4.         1];
5.     [n, m] = size(A);
6.     B = A;
7.     %% 标准化
8.     Z = B ./ repmat(sum(B.*B).^0.5, n, 1);
9.     disp("标准化后的矩阵 Z =")
10.    disp(Z)
11.    %% 计算优劣解距离
12.    D_P=sum([(Z-repmat(max(Z),n,1)).^2],2).^0.5 %距离 D+
13.    D_N=sum([(Z-repmat(min(Z),n,1)).^2],2).^0.5 %距离 D-

```

```
14. disp("正理想距离")
15. disp(D_P)
16. disp("负理想距离")
17. disp(D_N)
18.
19. %% 相对接近度
20. S=D_N./(D_P+D_N) %各自的相对接近度
21. disp(S)
22. plot(1:3,S)
```

9 九、参考文献

-
- [1] 董灵波. 基于模拟退火算法的森林多目标经营规划模拟[D].东北林业大学,2016.DOI:10.27009/d.cnki.gdblu.2016.000042.
 - [2] 李芳,李东坪.基于熵权法的组合评价模型[J].信息技术与信息化,2021(09):148-150.
 - [3] 吴振,吴红兰.基于改进遗传算法的无人机航路规划[J].电子测量技术,2021,44(24):52-58.DOI:10.19651/j.cnki.emt.2107877.