航运公司航线问题的数学模型

摘要

某航运公司承担了11个港口城市A-K的20条固定航线的32个航班运输任务。现已知各条航线的起点、终点和计划航班数，港口之间的航程距离以及船只最大连续航行能力。本文需要对该航运公司的经营进行优化。

针对问题一：首先，考虑到航班的最大连续航行能力，将各个港口之间超过15天的路线视为不存在；其次，将补给休整的1天算入前序航程；再次，利用最短路 Dijkstra 算法寻找20条航线的最短路径；最后，将得到的最短路径减去在目的地港口“接受补给”的1天，得到最终的最佳航线。20条航线的最佳航行天数为：36，27，26，27，27，43，14，15，22，11，20，15，6，41，24，15，43，41，31，12.

针对问题二：首先，将该航运公司的整个营运生命周期分为“开始营运-稳态”与“稳态”这两个阶段。“开始营运-稳态”阶段中各个港口投入的船只数量，等于“稳态”阶段系统内的全部船只数。其次，选定特征较易把握的“稳态”阶段，计算出各条航线上的周转船只总数为881；再将“稳态”时的船只调度问题转化为运筹学中的“运输问题”，利用整数规划，计算出各个港口之间的调度船只总数为46. 所以，航运公司共需船只927艘。此外，“稳态”阶段的调度方案为B→C，D→E，I→H.

针对问题三：首先，对无调运情况下各个港口的船只进出情况进行分析，给出了各港口出现稳定“余缺数”的时间点。其次，通过若干项原则与判断条件、以及多个方案的比较，获得了“开始营运-稳态”阶段的调运方案。再次，计算出“开始营运-稳态”阶段在第60天结束。然后，利用计算机程序模拟11个港口每天需要自备船只数量的情况。最后，将“开始营运-稳态”阶段调运方案，与“稳态”阶段调运方案进行汇总：港口I在第24天~第29天每天向港口B调运1艘，从第30天开始每天向港口H调运1艘；港口D从第33天开始每天向港口E调运1艘；港口B从第43天开始每天向港口C调运1艘。

针对问题四：首先，为简化模型，仅考虑“稳态”阶段的调度问题。其次，引入了“方向因子”，将航班进出港口的行为进行量化，高效地刻画了各个港口的航班“余缺数”问题。再次，进行整数规划，得到调整后的方案：20条航线的航班数量调整为为3，1，0，0，3，3，0，1，3，0，3，0，0，3，0，3，3，3，3，0. 此时总利润由调整前的4973万元上升至6090万元。

本文的特色在于，将该航运公司的整个营运周期纳入了考虑范围，并将其分为“开始营运-稳态”和“稳态”两个阶段，分别讨论，更加有力地把握了整个航运过程的变化，并利用程序对各个港口投入船只的过程进行了模拟。此外，“方向因子”的引入，更加高效地刻画了船只进出给各个港口带来的“余缺数”问题，使得整数规划变得简洁清晰。

关键词　运输问题；Dijkstra 算法；整数规划；动态模拟；调度方案

1. 问题的重述

某航运公司承担11个港口城市A-K的20条固定航线32个航班的物资运输任务。已知各条航线的起点、终点城市及每天航班数和各港口之间的航程距离（以连续航行天数计算）。

假设各航班的船只是同一型号的，载货时最大连续航行能力为15天，连续航行15天后需要停靠港口码头进行补给，休整1天后可继续航行。空载时连续航行天数为40天。

现有以下几个问题：

1、请给出各航班的最佳航线（航行天数最少）。

2、各航班装货和卸货各需要一天时间，为保证各航班正常运货需求，航运公司需配备的船只至少是多少？

3、某些港口城市需要的船只数量和到达的数量并不相等，请给出最佳的船只调运方案。

4、各航班航行完成后的收入按航行天数的15倍来计算（单位: 万元），满载和空载航行时，每天的花费大概分别是6万元和4万元。请问航运公司是否需要对航班进行调整（如取消某些航班）以提高收益？如果需要的话，请给出调整方案。

2. 问题的分析

2.1 问题（1）的分析

问题（1）要求给出各航班的最佳航线（航行天数最少）。

首先，考虑到航班的最大连续航行能力，将各个港口之间超过15天的路线视为不存在；其次，将补给休整的1天算入前序航程；再次，利用最短路Dijkstra算法寻找20条航线的最短路径；最后，将得到的最短路径减去在目的地港口“接受补给”的1天，得到最终的最佳航线。

此外，在问题（1）中还应依据上述方法，分析船只在各个港口之间空载调度时，最佳航程需要的天数。

2.2 问题（2）的分析

问题（2）要求计算在①航班装货、卸货各需一天，②航班正常营运得到满足这两个情况下，航运公司需配备的最少船只数量。

首先，将该航运公司的整个营运生命周期分为“开始营运-稳态”与“稳态”这两个阶段。并给出各个阶段的定义、具体特征的分析，以及这两个阶段之间的船只数量关系。其次，选定特征较易把握的“稳态”阶段，计算各条航线上的周转船只总数；再将“稳态”时的船只调度问题转化为运筹学中的“运输问题”，利用整数规划，计算出各个港口之间的调度船只总数。两者之和就是航运公司所需船只总数。

此外，整数规划模型还可以给出“稳态”阶段的船只调度方案。

2.3 问题（3）的分析

问题（3）要求给出在港口出现“缺船”和“余船”时，船只调运的最佳方

案。

首先，对无调运情况下各个港口的船只进出情况进行分析，给出了各港口出现稳定“余缺数”的时间点。其次，讨论“开始营运-稳态”阶段的船只调运方案。以“余船”港口为切入点，通过若干项原则与判断条件、以及多个方案的比较，寻找满足①调运时间最短，②需要船只最少的这两个条件的调运方案。再次，通过找出的最优调运方案，计算“开始营运-稳态”阶段结束的时间。然后，利用计算机程序模拟 11 个港口在“开始营运-稳态”阶段内，每天需要自备船只数量的情况，进行模型的验证。

最后，将第（3）问得出的“开始营运-稳态”阶段调运方案，与第（2）问得出的“稳态”阶段调运方案进行汇总，给出最终调运方案。

2.4 问题（4）的分析

问题（4）要求在考虑航班收益、调度成本的情况下，对各个航线的航班数量进行调整，以提高收益。

首先，为简化模型，仅考虑“稳态”阶段的调度问题。其次，针对第（4）问，增加了部分模型的假设，并定义了新的符号。这里引入了“方向因子”，将航班进出港口的行为进行量化，高效地刻画了各个港口带来的航班“余缺数”问题。再次，基于新的模型假设与符号，进行整数规划，得到调整后的方案。最后，从航班数量变化、调度方案变化、最终利润变化这三个方面，①将新的方案与原有方案进行对比，②分析新方案的实际变化与预计变化的差异。最终给出相应的结论。

3. 模型的假设与符号的说明

3.1 模型的假设

（1）各航班的船只是同一型号的，在各个港口、各条航线均可以正常使用；

（2）所有航线从第 0 天开始装货，第 1 天航班正式出发；

（3）对于到达港口或者离开港口的航班，开始装货即算作“离港”，卸货完毕才算“进港”；

（4）“进港”的船只可以立即“离港”，无需休整；

（5）所有的航程距离（天数）均为严格的整数；

（6）船只的“离港”“进港”均在每天正午 12：00 进行，文中的“第t天”指第t天正午 12：00；

（7）所有的到港船只不得闲置；

（8）航运公司的所有船只状况良好，无报废、退出营运系统的情况。

3.2 符号的说明

表 1　　符号的说明

符号	说明	单位
d_{ij}	港口i和港口j的距离	天
δ_i	第i条航线的中转次数	次
D_i	第i条航线的航程距离	天
d'_{ij}	港口i和港口j的调度距离	天
y_i	调整后第i条航线的航班数量	次/天
c_i	第i条航线上每次航班的收入	元/次
M_{ij}	从港口i到港口j的船只调运空载成本	元/艘
k_{ij}	从港口i向港口j的调运船只数量	艘/天
e_{ij}	航线和港口之间的“方向因子”	—

4. 模型的建立、求解与分析

4.1 问题（1）模型的建立、求解与分析

4.1.1 各航班最佳航线

根据模型的假定，共有 11 个港口城市的 20 条固定航线、32 个航班。由于载货时最大连续航行能力为 15 天，所以超过 15 天的航线必须停靠码头进行补给，休整 1 天后继续航行。本文拟采用 Dijkstra 最短路算法寻找最佳航线。

首先，由于船只载货的最大连续航行能力为 15 天，所以保留各港口 A-K 之间航程距离不超过 15 天的路线，并且删除超过 15 天的路线。

$$d_{ij}^0=\begin{cases}d_{ij}, & d_{ij}\le 15\\ 0, & d_{ij}>15\end{cases}$$

例如，港口 A 与港口 B 的距离为 11 天（≤15），将此路线保留；但是港口 A 与港口 C 的距离为 27 天（>15），将此路线删除。

其次，由于停靠码头进行补给需要消耗 1 天，所以将所有港口之间的航程距离加 1，将进行补给所消耗的天数算入前序航程。

$$d_{ij}^1=d_{ij}^0+1$$

例如，港口 A 与港口 B 的距离为 11 天，则此处算作 12 天。

然后，以d_{ij}^1为各条路径的权值，利用 Dijkstra 算法计算各个航线起点和终点的最短路径，得到D_i^0。当然，当船只到达终点时，不再需要补给和进行补给消耗的 1 天时间，所以最终的最短航线为$D_i = D_i^0 - 1$. 由于得到的最短路径减去 1 后仍为最短路径，所以该算法是可行的。

利用 MATLAB 编写程序，得到的最佳航线如下。

表 2　　最佳航线

航线	起点-终点	路线	航线距离D_i
1	A-F	A-E-G-F	36
2	B-C	B-E-G-C	27
3	C-D	C-G-E-D	26
4	C-B	C-G-E-B	27
5	D-F	D-E-G-F	27
6	D-H	D-I-K-H	43
7	E-I	E-I	14
8	E-G	E-G	15
9	F-I	F-G-I	22
10	F-J	F-J	11
11	F-K	F-G-K	20
12	G-E	G-E	15
13	G-C	G-C	6
14	H-B	H-K-I-B	41
15	H-G	H-K-G	24
16	I-K	I-K	15
17	J-A	J-G-E-A	43
18	K-A	K-G-E-A	41
19	K-D	K-I-D	31
20	K-G	K-G	12

4.1.2 各港口空载调度航线

当船只在各港口间空载调度时，连续航行天数为 40 天。观察各港口之间的航程距离，大于 40 天的仅有港口 A-港口 H。利用上述算法，求得港口 A-港口 H 之间的空载调度距离为 44 天，中途在港口 B 停靠休整。

所以，得到任意港口i和港口j的调度距离d'_{ij}.

$$d'_{ij} = \begin{cases} 44, & i = A, j = H \text{ 或 } i = H, j = A \\ d_{ij}, & \text{其他} \end{cases}$$

4.2 问题（2）模型的建立、求解与分析

4.2.1 “开始营运-稳态”阶段与“稳态”阶段

事实上，在假定 20 条航线的各个航班自始至终营运正常的情况下，航运公

司营运的生命周期可以分为“开始营运-稳态”和“稳态”这两个阶段：

“开始营运-稳态”阶段指，需要某些港口自备船只的阶段；

“稳态”阶段指，所有港口均不需要自备船只的阶段。

“稳态”阶段各个港口的航班情况统计如下。

表 3　“稳态”时各港口航班统计情况

港口	每天进港航班数	每天出港航班数	航班余缺数
A	2	2	0
B	3	2	1
C	3	4	-1
D	5	4	1
E	2	3	-1
F	4	4	0
G	3	3	0
H	2	3	-1
I	3	2	1
J	1	1	0
K	4	4	0

此时有 5 个港口进港出港航班平衡，3 个港口每天多一次航班，3 个港口每天少一次航班。为保证各个港口船只的供需平衡，需要在 6 个不平衡的港口之间调度，使得最终每个港口每天进港的船只最终全部离港，而需要离港的船只全部来源于进港船只——从而各个港口无需自备船只。

然而，在航运公司刚刚开始营运的阶段（如营运第 1 天，第 2 天…），每个港口每天必须保证正常数量的航班离港，却因为应进港的船只还在海上，而无法收到相应的进港船只。此时，为了满足各个航班的正常营运，各个港口必须自备船只。

例如，对于港口 A，每天应满足 2 个班次的航班开始装货、准备离港，但从航线装货的第 0 天算起，43 天（装货 1 天+航线距离D_{18}天+卸货 1 天）后，才能通过航线 18 收到可以使用的 1 艘船只。所以，在前 43 天（第 0 天至第 42 天），港口 A 就不得不自备 2*43=86 条船只。第 43 天、第 44 天，仍然需要自备 1 艘船只。直到第 45 天，除了航线 18 的 1 艘船只，港口 A 还收到了航线 17 的 1 艘船只。至此，港口 A 才出现了稳定的余缺数，并且余缺数为 0，故不再需要自备船只。

事实上，纵观航运公司的整个营运生命周期，在“开始营运-稳态”阶段中所投入的船只，全部进入了营运系统，且后续并不存在船只报废、退出系统的情况。所以，可以认为，先前各个港口投入的船只，在数量上等同于“稳态”阶段航运公司拥有的全部船只。

4.2.2 航运公司配备的最少船只数量

由于“稳态”阶段的规律更易把握，所以在讨论航运公司配备的船只数量时，就以“稳态”阶段的情况为依据进行计算。

Step 1. 计算载货航程需要的周转船只数。

根据模型的假设，对于到达港口或离开港口的航班，开始装货即算作“离港”，卸货完毕才算“进港”。所以，将航班装货所需的 1 天时间、卸货所需的 1 天时

间算入航程，得到各航线的“实际航程”D_i^1（天）.

$$D_i^1=1+D_i+1$$

那么，在各航线上参与航班营运的所有船只数，即“周转船只数”为

$$航线i周转船只数=D_i^1\times 航线i每天航班数$$

表 4　各航线周转船只数

航线	起点-终点	装货	航线距离D_i	卸货	D_i^1	航班数	周转船只数
1	A-F	1	36	1	38	2	76
2	B-C	1	27	1	29	2	58
3	C-D	1	26	1	28	3	84
4	C-B	1	27	1	29	1	29
5	D-F	1	27	1	29	2	58
6	D-H	1	43	1	45	2	90
7	E-I	1	14	1	16	2	32
8	E-G	1	15	1	17	1	17
9	F-I	1	22	1	24	1	24
10	F-J	1	11	1	13	1	13
11	F-K	1	20	1	22	2	44
12	G-E	1	15	1	17	2	34
13	G-C	1	6	1	8	1	8
14	H-B	1	41	1	43	2	86
15	H-G	1	24	1	26	1	26
16	I-K	1	15	1	17	2	34
17	J-A	1	43	1	45	1	45
18	K-A	1	41	1	43	1	43
19	K-D	1	31	1	33	2	66
20	K-G	1	12	1	14	1	14
			合计				881

最终得到，20 条航线的“周转船只数”共有 881 艘。

Step 2. 计算各港口间调度所需船只数

由表 3 可知，“稳态”时存在航班“余船”的港口为 B、D 和 I，存在航班“缺船”的港口为 C、E 和 H。利用运筹学中的“运输问题”，寻求供需平衡。

表 5　“余船”港口与“缺船”港口的距离

港口间空载调度距离d'_{ij}	C	E	H	每日“余船”
B	18	4	32	1
D	18	3	32	1
I	15	14	25	1
每日“缺船”	1	1	1	

设：

每天由 B 调往 C、E 和 H 的船只数量为x_{11}、x_{12}和x_{13}；

每天由 D 调往 C、E 和 H 的船只数量为x_{21}、x_{22}和x_{23}；

每天由 I 调往 C、E 和 H 的船只数量为x_{31}、x_{32}和x_{33}.

为了使得各港口间调度所需船只总数最少，进行整数规划：

$$
\min \quad \begin{aligned} Z = 18x_{11} + \quad 4x_{12} + 32x_{13} \\ +18x_{21} + \quad 3x_{21} + 32x_{23} \\ +15x_{31} + 14x_{32} + 25x_{33} \end{aligned}
$$

$$
\text{s.t} \begin{cases} x_{11} + x_{12} + x_{13} = 1 \\ x_{21} + x_{22} + x_{23} = 1 \\ x_{31} + x_{32} + x_{33} = 1 \\ x_{11} + x_{21} + x_{31} = 1 \\ x_{12} + x_{22} + x_{32} = 1 \\ x_{13} + x_{23} + x_{33} = 1 \\ \quad x_{ij} \geq 0 \\ \quad x_{ij} \in Z \end{cases} .
$$

利用 LINGO 进行求解，得到结果如下。

表 6　　调度所需船只数量的优化结果

变量	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}	Z
取值	1	0	0	0	1	0	0	0	1	46

最终得到，“稳态”阶段的调度方案为：

港口 B→港口 C，每天 1 艘，每艘 18 天到达；

港口 D→港口 E，每天 1 艘，每艘 3 天到达；

港口 I→港口 H，每天 1 艘，每艘 25 天到达。

一共需要调度船只Z=46 艘。

Step 3. 计算船只总数

$$
\text{船只总数} = \text{周转船只总数} + \text{调度船只总数}
$$

所以，该航运公司一共需要配备 927 艘船只。

4.3 问题（3）模型的建立、求解与分析

4.3.1 无调运情况下各港口的船只进出分析

事实上，船只在港口之间的调运，不仅仅局限于上述整数规划得出的、“稳态”阶段的调运。在到达稳态之前的“开始营运-稳态”阶段，同样需要船只的调运。

为了后续对“开始营运-稳态”阶段的船只调运情况进行分析，首先需要分析出，在没有调运的情况下，各港口从开始营运的第 0 天到“余缺数”出现稳定期间的船只变化情况。

依据模型的假设（2），所有航线从第 0 天开始装货，第 1 天航班正式出发。

1. **A 港口，每天离港航班 2 艘**

   第 0~42 天，每天需要自备 2 艘；

   第 43 天，由航线 18 收到 1 艘，需要自备 1 艘；

   第 44 天，由航线 18 收到 1 艘，需要自备 1 艘；

   第 45 天，由航线 18 收到 1 艘，由航线 17 收到 1 艘，此后每天既不余船、也不缺船，“余缺数”稳定。

2. **B 港口，每天离港航班 2 艘**

   第 0~28 天，每天需要自备 2 艘；

第 29 天，由航线 4 收到 1 艘，需要自备 1 艘；
…
第 42 天，由航线 4 收到 1 艘，需要自备 1 艘；
第 43 天，由航线 4 收到 1 艘，由航线 14 收到 2 艘，此后每天余 1 艘船，“余缺数”稳定。

3. **C 港口，每天离港航班 4 艘**
第 0~7 天，每天需要自备 4 艘；
第 8 天，由航线 13 收到 1 艘，需要自备 3 艘；
…
第 28 天，由航线 13 收到 1 艘，需要自备 3 艘；
第 29 天，由航线 13 收到 1 艘，由航线 2 收到 2 艘，此后每天缺 1 艘船，“余缺数”稳定。

4. **D 港口，每天离港航班 4 艘**
第 0~27 天，每天需要自备 4 艘；
第 28 天，由航线 3 收到 3 艘，需要自备 1 艘；
…
第 32 天，由航线 3 收到 3 艘，需要自备 1 艘；
第 33 天，由航线 3 收到 3 艘，由航线 19 收到 2 艘，此后每天余 1 艘船，“余缺数”稳定。

5. **E 港口，每天离港航班 3 艘**
第 0~16 天，每天需要自备 3 艘；
第 17 天，由航线 12 收到 2 艘，此后每天缺 1 艘船，“余缺数”稳定。

6. **F 港口，每天离港航班 4 艘**
第 0~28 天，每天需要自备 4 艘；
第 29 天，由航线 5 收到 2 艘，需要自备 2 艘；
…
第 37 天，由航线 5 收到 2 艘，需要自备 2 艘；
第 38 天，由航线 5 收到 2 艘，由航线 1 收到 2 艘，此后每天既不余船、也不缺船，“余缺数”稳定。

7. **G 港口，每天离港航班 3 艘**
第 0~13 天，每天需要自备 3 艘；
第 14 天，由航线 20 收到 1 艘，需要自备 2 艘；
…
第 16 天，由航线 20 收到 1 艘，需要自备 2 艘；
第 17 天，由航线 20 收到 1 艘，由航线 8 收到 1 艘，需要自备 1 艘；
…
第 25 天，由航线 20 收到 1 艘，由航线 8 收到 1 艘，需要自备 1 艘；
第 26 天，由航线 20 收到 1 艘，由航线 8 收到 1 艘，由航线 15 收到 1 艘，此后每天既不余船、也不缺船，“余缺数”稳定。

8. **H 港口，每天离港航班 3 艘**
第 0~44 天，每天需要自备 3 艘；
第 45 天，由航线 6 收到 2 艘，此后每天缺 1 艘船，“余缺数”稳定。

9. **I 港口，每天离港航班 2 艘**
第 0~15 天，每天需要自备 2 艘；

第 16 天，由航线 7 收到 2 艘，“余缺数”暂时为 0；

…

第 23 天，由航线 7 收到 2 艘，“余缺数”暂时为 0；

第 24 天，由航线 7 收到 2 艘，由航线 9 收到 1 艘，此后每天多 1 艘船，“余缺数”稳定

10. J 港口，每天离港航班 1 艘

第 0~12 天，每天需要自备 1 艘；

第 13 天，由航线 10 收到 1 艘，此后每天既不余船、也不缺船，“余缺数”稳定。

11. K 港口，每天离港航班 4 艘

第 0~16 天，每天需要自备 4 艘；

第 17 天，由航线 16 收到 2 艘，需要自备 2 艘；

…

第 21 天，由航线 16 收到 2 艘，需要自备 2 艘；

第 22 天，由航线 16 收到 2 艘，由航线 11 收到 2 艘，此后每天既不余船、也不缺船，“余缺数”稳定。

需要注意的是，无论是否在各个港口间进行调运，各港口出现“余缺数”稳定的时间是固定的。而港口间的调运行为仅影响需要准备的船只数量。

可以看出，对于出现余船的 B、D 和 I 港口，港口 I 率先开始出现余船，耗时 24 天；之后，港口 D 开始出现余船，耗时 33 天；港口 B 最后开始出现余船，耗时 43 天。

4.3.2 “开始营运-稳态”阶段的船只调运方案

在调运船只时，需要明确以下几个原则：

（1）当某一港口出现“余船”时，才开始调运；

（2）调运时，选择离“余船”港口最近的“缺船”港口。之所以这样选择，是由于从“余船”港口往“缺船”港口调船时，调运船只航行天数越长，“缺船”港口需自备船只的天数就越多，从而需自备的船只就越多，所花费的成本就越高。

（3）若存在多个“余船”港口，当离他们最近的“缺船”港口均为同一港口时，采用多个方案相互比较的方法，选取各港口调运距离之和最短的方案；

（4）将多个港口按照开始出现“余船”的先后顺序排列，即港口 I、港口 D、港口 B。

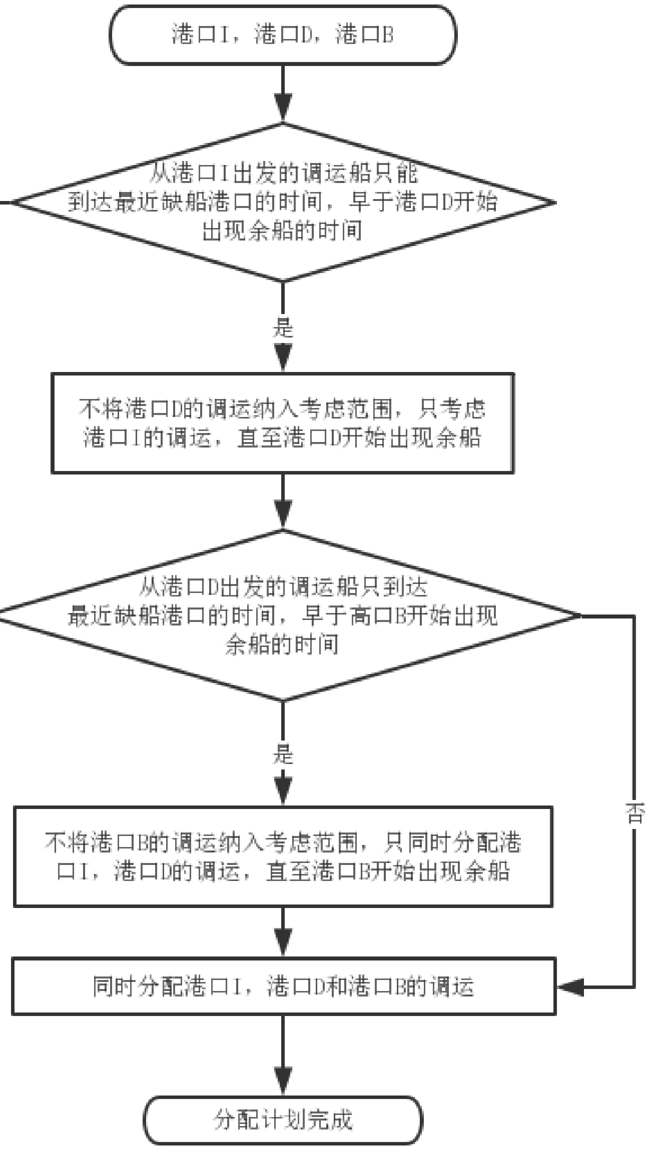

图 1　　调运方案分配流程图

依据上述原则和流程，可以得到最佳调运方案。

Step 1. **判断条件 1**

港口 I 从第 24 天开始出现 1 艘余船。该艘余船不应闲置，必须调运至其他港口。当船到达目的地港口时，还缺少船的目的地港口为 B、C、E、H，到达时间分别为第 37 天、第 39 天、第 38 天、第 49 天.

港口 D 从第 33 天开始出现 1 艘余船。该艘余船不应闲置，必须调运至其他港口。当船到达目的地港口时，还缺少船的目的地港口为 A、B、C、E、H，到达时间分别为第 44 天、第 36 天、第 51 天、第 35 天、第 65 天。

显然，距离港口 I 最近的“缺船”港口为 B，达到 B 的时间晚于港口 D 开始出现余船的时间。

Step 2. 判断条件 2

港口 B 从第 43 天开始出现 1 艘余船。该艘余船不应被闲置，必须调运至其

他港口。当船到达目的地港口时，还缺少船的目的地港口为 C、E、H，到达时间分别为第 61 天、第 47 天、第 75 天。

显然，距离港口 D 最近的“缺船”港口为 B 和 E，到达 B 和 E 的时间早于港口 B 开始出现余船的时间。

Step 3. 第一次调运分配

不将“余船”港口 B 的调运纳入考虑范围。从第 24 天开始，同时分配港口 I 和港口 D 的调运，直至港口 B 开始出现余船的第 43 天前夕（即第 42 天）。

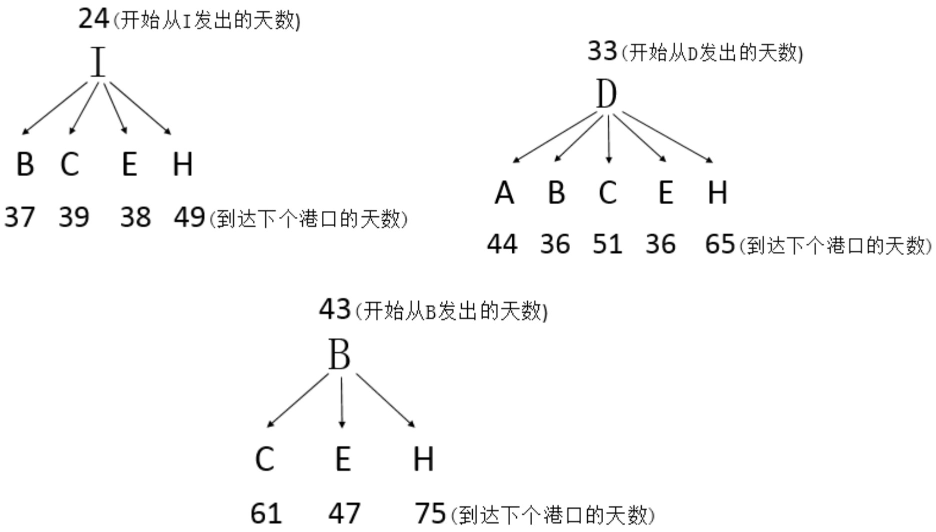

图 2　调运方案的分配（1）

由于距离港口 I 和港口 D 的最近“缺船”港口都包括 B，但是港口 B 在第 36 天~第 42 天期间，仅缺少 1 艘船，只能接受 1 个港口的调度。所以，必须通过多个方案比较的方法，确定最优方案。

好在，当选择调度路线 I→B（即不能再选择 D→B）时，距离港口 D 最近的“缺船”港口为 E；而当选择调度路线 I→E（即不能再选择 D→E）时，距离港口 D 最近的“缺船”港口为 B. 无论如何，最终选择的两个“缺船”港口一定是 B 和 E，而这两个港口在第 36 天~第 42 天期间又每天都缺 1 艘船。综上，只需要比较两个方案的调度路线距离之和即可。

表 7　调运方案的比较（1）

	方案 1		方案 2	
调度路线	I→B	D→E	I→E	D→B
距离	13	3	14	3
合计	16		17	

显然，I→B，D→E 的调运方案更为合理。

其中，I→B 的调运过程从第 24 天开始，连续调运 6 天，每天调运 1 艘；这样，由于d'_{IB}=13，到第 42 天，B 港口可以节省 6 艘原本需要自备的船只。D→E 的调运过程从第 33 天开始，连续调运 7 天，每天调运 1 艘；这样，由于d'_{DE}=3，到第 42 天前夕，E 港口可以节省 7 艘原本需要自备的船只。

Step 4. 第二次调运分配

同时分配港口 I、港口 D 和港口 B 的调运。从第 43 天开始，直至“开始营运-稳态”阶段的结束。

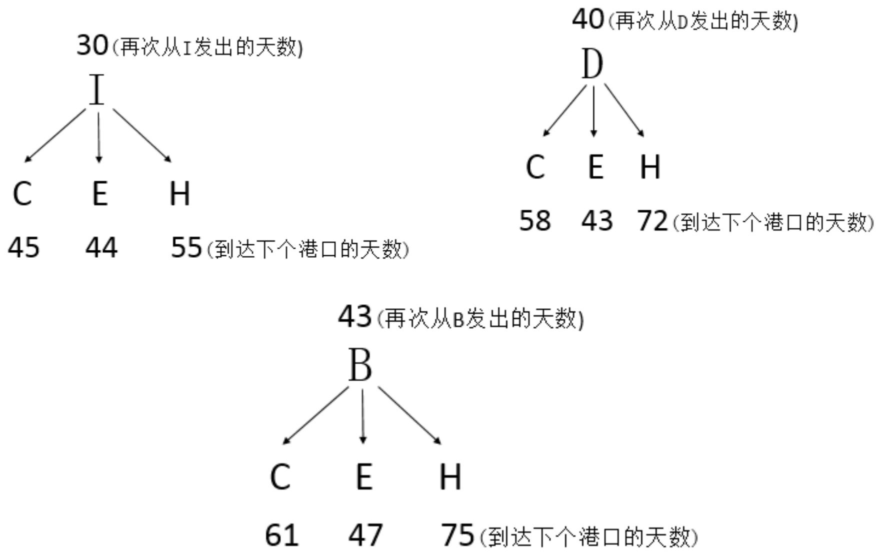

图 3　　调运方案的分配（2）

注意到，此时，与“余船”港口 I、D、B 相对应的“缺船”港口仅剩下 C、E、H；其他的港口都已经进入既不缺船也不余船的“余缺数”恒定状态。而港口 C、E、H 也恰好就是表 3 中“稳态”时航班“余缺数”为-1 的 3 个港口。因此，有理由相信，第二次调运分配的结果与“稳态”时调运分配的结果相同。

表 8　　调运方案的比较（2）

编号	调度路线	距离和	编号	调度路线	距离和
1	I→C，D→E，B→H	50	4	I→E，D→H，B→C	64
2	I→C，D→H，B→E	51	5	I→H，D→C，B→E	47
3	I→E，D→C，B→H	64	6	I→H，D→E，B→C	46

通过表 8 中各个调运方案的比较，调运方案 6 更为合理。而这也证明了我们的先前的猜测——第二次调运分配方案与“稳态”时方案相同，参与的调运船只数也相同，都为 46。

最后，可以计算出“开始营运-稳态”阶段结束的时间。

对于调运路线 I→H，从第 55 天开始，港口 H 每天收到 1 艘来自港口 I 的调运船只；此后，港口 H 无需再自备船只；

对于调运路线 D→E，从第 43 天开始，港口 E 每天收到 1 艘来自港口 D 的调运船只；事实上，港口 E 在第一次调运分配的过程中，从第 36 天开始，就已经无需再自备船只；

对于调运路线 B→C，从第 61 天开始，港口 C 每天收到 1 艘来自港口 B 的调运船只；此后，港口 C 无需再自备船只。

综上所述，第 60 天是“开始营运-稳态”阶段的最后一天。从第 61 天开始，所有港口都无需再自备船只，每天进港的船只全部出港，出港的船只全部来源于当天进港船只，该航运公司的营运达到“稳态”。

4.3.3 方案的验证

根据 4.2.2，在“开始营运-稳态”阶段的调运方案为：

表 9　“开始营运-稳态”阶段的调运方案

余船港口	发船日期	缺船港口	收船日期
I	第 24 天~第 29 天	B	第 37 天~第 42 天
	第 30 天~第 48 天	H	第 55 天~第 61 天
D	第 33 天~第 58 天	E	第 36 天~第 61 天
B	第 43 天	C	第 61 天

利用 C++编写程序进行模拟，模拟的基本流程与规则为：

（1）从第 0 天开始模拟；

（2）选定模拟的港口i；

（3）港口i正常的进港出港航班都得到满足；

（4）按照表 9 中的方案进行各港口之间船只的调运；

（5）对于港口i的第t天，将正常进港的航班数记为a_{it}，正常出港的航班数记为b_{it}，调入的船只数记为c_{it}，调出的船只数记为d_{it}；

（6）则港口i的第t天需自备的船只数为$b_{it}+d_{it}-a_{it}-c_{it}$；

（7）则港口i在“开始营运-稳态”阶段需自备的累计船只数为

$$\sum_{t=0}^{60}(b_{it}+d_{it}-a_{it}-c_{it}).$$

从而得到了各个港口在开始营运前需要准备的船只数：

表 10　各个港口需要准备的船只数量

港口	A	B	C	D	E	F
数量（艘）	88	66	127	117	70	134
港口	G	H	I	J	K	合计
数量（艘）	57	145	32	13	78	927

各个港口需要准备的船只数量之和，与问题（2）中得到的“周转船只总数”与“调度船只总数”之和相互吻合，也从侧面印证了结果的合理性。

此外，附录 A.3.2 中给出了从第 0 天至第 61 天，各个港口每天需要自备的船只数量，以及累计数量的具体数据。

4.3.4 最终调运方案

最后，将“开始营运-稳态”阶段与“稳态”阶段的调运方案进行汇总，得到最终的调运方案。

表 11　最终调运方案

余船港口	发船日期	缺船港口	收船日期
I	第 24 天~第 29 天	B	第 37 天~第 42 天
	第 30 天~	H	第 55 天~
D	第 33 天~	E	第 36 天~
B	第 43 天~	C	第 61 天~

4.4 问题（4）模型的建立、求解与分析

4.4.1 模型的建立

问题（4）要求对航线进行调整，以提高收益。在这一问中，为简化模型，不考虑“开始营运-稳态”阶段的调度问题，而仅考虑达到“稳态”阶段后的调度问题。

根据题目所给条件，船只满载时，第i条航线的利润为$15d_{ij}-6(D_i-\delta_i)$元；而船只在港口之间调度时，从港口i到港口j的成本为$4\times d'_{ij}$元。若想要尽可能提高收益，预计的调整为：

（1）增加航程距离较长的航线的航班数量；

（2）减少航程距离较短的航线的航班数量；

（3）减少港口之间调度距离较长的调度行为。

事实上，无论各航线航班数量如何调整，都应该满足以下条件：

（1）不新增航线，但可能取消航线；

（2）既有航线上的航班数量可增可减，当数量减少至 0 时，意味着该条航线被取消；

（3）各个港口无需自备船只，全部达到“稳态”。即每日进港的船只（包括正常进入的航班和调运进入该港口的船只）都在当日离开港口，且每日离港的船只（包括正常离开的航班和调运离开该港口的船只）都来源于当日进港的船只；

（4）某一条航线上的航班不能过多。这是由于，一方面，各个港口有限的吞吐量可能无法承担过多的航班；另一方面，当一条航线上的航班数量过多时，航运市场接近饱和，多余的航班必将空载，导致预计的利润变成亏损。

这里，不妨假设每条航线的航班最多为 3 班次/天。

（5）该航运公司所有航线上的航班数量之和不应过多。这一方面是条件（4）衍生而得；另一方面是由于过多的航班需要过多的航船，而航船的购置与维护费用也是一笔不小的开支。

这里，不妨假设所有航线上的航班数量之和不超过现行数量，即 32 班次/天。

综上，设变量：

y_i为调整后第i条航线的航班数量，单位为“次/天”；

c_i为第i条航线上每次航班的收入，$c_i=15d_{ij}-6(D_i-\delta_i)$，单位为“元/次”；

M_{ij}为从港口i到港口j调运 1 艘船只的空载成本，$M_{ij}=4d'_{ij}$，单位为“元/艘”；

k_{ij}为每天从港口i向港口j调运的船只数量，单位为“艘/天”；

所以，总收入为$\sum_{i=1}^{20} y_i c_i$，单位为“元/天”；

总成本为$\sum_{i=1}^{11}\sum_{i=1}^{11} k_{ij}M_{ij}$，单位为“元/天”；

总利润为$\mathrm{Profit}=\sum_{i=1}^{20} y_i c_i-\sum_{i=1}^{11}\sum_{i=1}^{11} k_{ij}M_{ij}$，单位为“元/天”。

此外，定义“方向因子”e_{ij}，下标i表示航线i，下表j表示港口j，

$$e_{ij}=\begin{cases}1, & \text{航线}i\text{的终点为港口}j\\ -1, & \text{航线}i\text{的起点为港口}j\\ 0, & \text{航线}i\text{的起点、终点均不是港口}j\end{cases}.$$

例如，对于第 1 条航线，终点为港口 F，那么$e_{1F}=1$；起点为港口 A，那么$e_{1A}=-1$；航线 1 的起点、终点均不是港口 B，那么$e_{1B}=0$.

然后，进行整数规划。

$$\max \text{Profit}=\sum_{i=1}^{20} y_i c_i-\sum_{i=1}^{11}\sum_{i=1}^{11} k_{ij} M_{ij}$$

$$\text{s.t}\begin{cases}\sum_{i=1}^{20} y_i e_{ij}+\sum_{i=1}^{11} k_{ij}-\sum_{i=1}^{11} k_{ji}=0\\ \sum_{i=1}^{11} y_i \leq 32\\ 0\leq y_i\leq 3\\ k_{ij}\geq 0\\ y_i, k_{ij}\in Z\end{cases}.$$

4.4.2 模型结果分析

对于上述模型，利用 LINGO 求解得到结果。

结果可以分为 3 部分进行讨论。

Step 1. 调整后航班数量y_i的讨论

表 12 调整后结果

航线	调整前班次	调整后班次y_i	港口距离d_{ij}	航线	调整前班次	调整后班次y_i	港口距离d_{ij}
1	2	3	29	11	2	3	17
2	2	1	18	12	2	0	15
3	3	0	18	13	1	0	6
4	1	0	18	14	2	3	32
5	2	3	19	15	1	0	17
6	2	3	32	16	2	3	15
7	2	0	14	17	1	3	36
8	1	1	15	18	1	3	34
9	1	3	20	19	2	3	25
10	1	0	11	20	1	0	12
合计				—	32	32	—

观察表 12，不难发现：

（1）经过调整，港口距离d_{ij}较大的航线航班次数明显增加；而d_{ij}较小的航

线航班次数明显减少，有的甚至于降为 0，使该航线不复存在。这显然符合常理，因为在满足各个港口达到“稳态”的情况下，港口距离d_{ij}越大的航班越赚钱；

（2）调整后，各条航线上的航班数量均不超过 3，而存在多条航线上的航班数量恰好为 3. 这说明“每条航线的航班最多为 3 班次/天”是必要的约束；

（3）调整后，所有航线上的航班数量之和仍为 32，说明“所有航线上的航班数量之和不超过 32 班次/天”是必要的约束。

Step 2. 调整后港口间船只调运方案k_{ij}的讨论

调整前，港口间调运方案为：

每天从港口 B 向港口 C 调运 1 艘船，调度距离为 18 天；

从港口 D 向港口 E 调运 1 艘船，调度距离为 3 天；

从港口 I 向港口 H 调运 1 艘船，调度距离为 25 天；

调整后，港口间调运方案为：

每天从港口 A 向港口 D 调运 3 艘船，调度距离为 11 天；

从港口 B 向港口 E 调运 1 艘船，调度距离为 4 天；

从港口 B 向港口 G 调运 1 艘船，调度距离为 16 天；

从港口 C 向港口 J 调运 1 艘船，调度距离为 11 天；

从港口 G 向港口 J 调运 2 艘船，调度距离为 14 天；

与预判不符的是，与先前相比，调运船只的数量增加了，所以调度的航行距离显著增加。这样的增加会导致船只空载调度的成本增加，显然不利于提高利润。出现这样的结果，或许是由于各个港口必须满足“稳态”条件的限制造成的。

Step 3. 调整后航运公司营运利润$\sum_{i=1}^{20} y_i c_i - \sum_{i=1}^{11}\sum_{i=1}^{11} k_{ij} M_{ij}$的讨论

表 13　调整前后收益对比

	航线收益	调运成本	总利润
调整前	5157	184	4973
调整后	6438	348	6090
调整后-调整前	1281	164	1117

可以看到，相较于调整前的方案：

（1）调整后的方案中，通过营运航线而得到的直接收益显著增加，这也印证了在 Step 1. 中的分析，即通过增加港口距离d_{ij}较大的航线航班次数、减小港口距离d_{ij}较小的航线航班次数能够增加收入；

（2）调整后的方案中，在各个港口之间调运船只花费的成本亦显著增加，这也印证了在 Step 2. 中的分析，即增加了船只调运的数量、调运航行距离，从而导致船只空载调度成本增加，不利于提高利润；

（3）综合考虑，在调整后的方案中，由于Δ航线收益 $>$ Δ调运成本，故

$$\Delta\text{总利润} = \Delta\text{航线收益} - \Delta\text{调运成本} > 0.$$

虽然“调运成本”有所增加，但这显然不能掩盖“航线收益”更为显著的增长。最终，调整后的方案比调整前的方案多出了 1117 万元的收益，在保证了各

个港口达到“稳态”的前提下，是更优的方案。

5. 模型的评价与改进方向

5.1 模型的优点

（1）利用最短路 Dijkstra 算法得到了各条航线的最短航程，思路清晰，算法简洁，避免了规划的冗杂；

（2）将该航运公司的整个营运生命周期分为“开始营运-稳态”阶段与“稳态”阶段，并且准确地辨别了两个阶段的区别与临界点，更加有力地把握了整个航运过程的变化；

（3）对于船只调运的分析，不仅仅局限于“稳态”阶段，还给出了“开始营运-稳态”阶段的方案；

（4）系统全面地分析了各个港口的船只出入情况，并且模拟了第 0 天~第 61 天各港口需自备船只的情况；

（5）引入“方向因子”，高效地刻画了船只进出给各个港口带来的“余缺数”问题。

5.2 模型的缺点及改进方向

（1）没有定量地考虑航运市场的饱和问题、各航班满载率的问题、港口吞吐量的上限问题以及船只购置与维护费用的问题。将这些因素纳入考虑，是模型改进的方向。

（2）缺乏对某些参数的灵敏度分析。例如，当航班的收益、船只调度费用等出现变化时，讨论模型的结果是否出现大的变动，是模型需要进一步讨论的方向。

6. 参考文献

[1] 姜启源，谢金星，叶俊. 数学模型（第四版）. 北京：高等教育出版社，2011.

[2] 胡运权，郭辉煌. 运筹学教程（第四版）. 北京：清华大学出版社，2012.

[3] 司守奎，孙玺菁. 数学建模算法与应用. 北京：国防工业出版社，2011.

[4] 袁新生，邵大宏，郁时炼. LINGO 和 Excel 在数学建模中的应用. 北京：科学出版社，2007.

附录

A.1 问题（1）的附录

Dijkstra 最短路算法的 MATLAB 代码

hangxian.m

```
clc, clear
a=zeros(11,11);
% a(1,2)=11;a(1,3)=27;a(1,4)=11;a(1,5)=12;a(1,6)=29;a(1,7)=26;a(1,8)=42;a(1,9)=22;a(1,10)=36;a(1,11)=34;
% a(2,3)=18;a(2,4)=3;a(2,5)=4;a(2,6)=19;a(2,7)=16;a(2,8)=32;a(2,9)=13;a(2,10)=28;a(2,11)=25;
% a(3,4)=18;a(3,5)=16;a(3,6)=9;a(3,7)=6;a(3,8)=17;a(3,9)=15;a(3,10)=11;a(3,11)=13;
% a(4,5)=3;a(4,6)=19;a(4,7)=16;a(4,8)=32;a(4,9)=15;a(4,10)=27;a(4,11)=25;
% a(5,6)=17;a(5,7)=15;a(5,8)=31;a(5,9)=14;a(5,10)=26;a(5,11)=24;
% a(6,7)=7;a(6,8)=17;a(6,9)=20;a(6,10)=11;a(6,11)=17;
% a(7,8)=17;a(7,9)=14;a(7,10)=14;a(7,11)=12;
% a(8,9)=25;a(8,10)=13;a(8,11)=11;
% a(9,10)=26;a(9,11)=15;
% a(10,11)=18;
a(1,2)=11;;a(1,4)=11;a(1,5)=12;
a(2,4)=3;a(2,5)=4;a(2,9)=13;
a(3,6)=9;a(3,7)=6;a(3,9)=15;a(3,10)=11;a(3,11)=13;
a(4,5)=3;a(4,9)=15;
a(5,7)=15;a(5,9)=14;
a(6,7)=7;a(6,10)=11;
a(7,9)=14;a(7,10)=14;a(7,11)=12;
a(8,10)=13;a(8,11)=11;
a(9,11)=15;
for i=1:11
    for j=1:11
        if(a(i,j)~=0)
            a(i,j)=a(i,j)+1;
        end
    end
end

a=a' ; %matlab 工具箱要求数据是下三角矩阵
```

[i,j,v]=find(a);

c=sparse(a)%构造稀疏矩阵，只储存非零元素及其位置，节约储存空间。用b=full(b)可转化为满阵

[x,y,z]=graphshortestpath(c,11,7,'Directed',false) % Directed 是标志图为有向或无向的属性，该图是无向图，对应的属性值为 false，或 0。

% h=view(biograph(c,[],'ShowArrows','off','ShowWeights','on'))

A.2 问题（2）的附录

LINGO 求解调度船只数的代码

船只数量.lg4

```
model:
min = 18*x11 + 2*x12 + 32*x13
      + 18*x21 + 3*x22 + 32*x23
      + 15*x31 +14*x32 + 25*x33;
!每日供船;
x11+x12+x12<=1;
x21+x22+x23<=1;
x31+x32+x33<=1;
!每日需船;
x11+x21+x31=1;
x12+x22+x32=1;
x13+x23+x33=1;
end
```

A.3 问题（3）的附录

A.3.1 模拟各港口每日自备船只数量的 C++代码

航线.cpp

```
#include<iostream>
#include<sstream>
using namespace std;
int ship[11],mi[11];
int AtoF=36, BtoC=27, CtoD=26, CtoB=27, DtoF=27,
DtoH=43, EtoI=14, EtoG=15, FtoI=22, FtoJ=11,
FtoK=20, GtoE=15, GtoC=6, HtoB=41, HtoG=24,
ItoK=15, JtoA=43, KtoA=41, KtoD=31, KtoG=12;


int ItoB_=13,DtoE_=3,ItoH_=25,BtoC_=18;
void fun(int t){
```

```
t-=1;

if(t<=KtoA)
	ship[0]-=2;
else if(t<=JtoA)
	ship[0]-=1;
else;

if(t<=CtoB)
	ship[1]-=2;
else if(t<=HtoB)
	ship[1]-=1;
else ship[1]++;

if(t<=GtoC)
	ship[2]-=4;
else if(t<=BtoC)
	ship[2]-=3;
else ship[2]-=1;

if(t<=CtoD)
	ship[3]-=4;
else if(t<=KtoD)
	ship[3]-=1;
else ship[3]++;

if(t<=GtoE)
	ship[4]-=3;
else ship[4]--;

if(t<=DtoF)
	ship[5]-=4;
else if(t<=AtoF)
	ship[5]-=2;
else;

if(t<=KtoG)
	ship[6]-=3;
else if(t<=EtoG)
	ship[6]-=2;
else if(t<=HtoG)
	ship[6]--;
else ;
```

```
	if(t<=DtoH)
		ship[7]-=3;
	else ship[7]--;

	if(t<=EtoI)
		ship[8]-=2;
	else if(t<=FtoI)
		ship[8]=ship[8];
	else ship[8]++;

	if(t<=FtoJ)
		ship[9]--;
	else;

	if(t<=ItoK)
		ship[10]-=4;
	else if(t<=FtoK)
		ship[10]-=2;
	else ;


}
void diaodu(int t){
	if(t>=24&&t<30) {   //I->B
		ship[8]--;
	}
	if(t>=(24+ItoB_) &&t<(30+ItoB_))
		ship[1]++;

	if(t>=33&&t<40){     //D->E
		ship[3]--;
	}
	if(t>=(33+DtoE_)&&t<(40+DtoE_)){     //D->E
		ship[4]++;
	}

	if(t>=30){     //I->H
		ship[8]--;
	}
	if(t>=(30+ItoH_)){     //I->H
		ship[7]++;
	}

	if(t>=40){     //D->E
```

```
        ship[3]--;
    }
    if(t>=(40+DtoE_)){      //D->E
        ship[4]++;
    }

    if(t>=43){      //B->C
        ship[1]--;
    }
    if(t>=(43+BtoC_)){      //B->C
        ship[2]++;
    }
}
void finmin(){
    for(int j=0;j<11;j++){
        if(mi[j]>ship[j])
            mi[j]=ship[j];
    }

}
int main(){
    for(int i=0;i<100;i++)
    {
//          for(int j=0;j<11;j++)
//              ship[j]=0;
        fun(i);
        diaodu(i);
        finmin();
        for(int j=0;j<11;j++)
            cout<<ship[j]<<" ";
        cout<<endl;
    }

    int n=0;
    for(int j=0;j<11;j++){
        char t = 'A'+j;
        cout<<t<<"最少船数为: "<<-mi[j]<<endl;
    }
    return 0;
}
```

A.3.2 模拟结果

表 14　各个港口每天需准备船只数量

天数	港口 A	港口 B	港口 C	港口 D	港口 E	港口 F	港口 G	港口 H	港口 I	港口 J	港口 K
0	2	2	4	4	3	4	3	3	2	1	4
1	2	2	4	4	3	4	3	3	2	1	4
2	2	2	4	4	3	4	3	3	2	1	4
3	2	2	4	4	3	4	3	3	2	1	4
4	2	2	4	4	3	4	3	3	2	1	4
5	2	2	4	4	3	4	3	3	2	1	4
6	2	2	4	4	3	4	3	3	2	1	4
7	2	2	4	4	3	4	3	3	2	1	4
8	2	2	3	4	3	4	3	3	2	1	4
9	2	2	3	4	3	4	3	3	2	1	4
10	2	2	3	4	3	4	3	3	2	1	4
11	2	2	3	4	3	4	3	3	2	1	4
12	2	2	3	4	3	4	3	3	2	1	4
13	2	2	3	4	3	4	3	3	2	0	4
14	2	2	3	4	3	4	2	3	2	0	4
15	2	2	3	4	3	4	2	3	2	0	4
16	2	2	3	4	3	4	2	3	0	0	4
17	2	2	3	4	1	4	1	3	0	0	2
18	2	2	3	4	1	4	1	3	0	0	2
19	2	2	3	4	1	4	1	3	0	0	2
20	2	2	3	4	1	4	1	3	0	0	2

21	2	2	3	4	1	4	1	3	0	0	2
22	2	2	3	4	1	4	1	3	0	0	0
23	2	2	3	4	1	4	1	3	0	0	0
24	2	2	3	4	1	4	1	3	0	0	0
25	2	2	3	4	1	4	1	3	0	0	0
26	2	2	3	4	1	4	0	3	0	0	0
27	2	2	3	4	1	4	0	3	0	0	0
28	2	2	3	1	1	4	0	3	0	0	0
29	2	1	1	1	1	2	0	3	0	0	0
30	2	1	1	1	1	2	0	3	0	0	0
31	2	1	1	1	1	2	0	3	0	0	0
32	2	1	1	1	1	2	0	3	0	0	0
33	2	1	1	0	1	2	0	3	0	0	0
34	2	1	1	0	1	2	0	3	0	0	0
35	2	1	1	0	1	2	0	3	0	0	0
36	2	1	1	0	0	2	0	3	0	0	0
37	2	0	1	0	0	2	0	3	0	0	0
38	2	0	1	0	0	0	0	3	0	0	0
39	2	0	1	0	0	0	0	3	0	0	0
40	2	0	1	0	0	0	0	3	0	0	0
41	2	0	1	0	0	0	0	3	0	0	0
42	2	0	1	0	0	0	0	3	0	0	0
43	1	0	1	0	0	0	0	3	0	0	0
44	1	0	1	0	0	0	0	3	0	0	0
45	0	0	1	0	0	0	0	1	0	0	0

46	0	0	1	0	0	0	0	1	0	0	0
47	0	0	1	0	0	0	0	1	0	0	0
48	0	0	1	0	0	0	0	1	0	0	0
49	0	0	1	0	0	0	0	1	0	0	0
50	0	0	1	0	0	0	0	1	0	0	0
51	0	0	1	0	0	0	0	1	0	0	0
52	0	0	1	0	0	0	0	1	0	0	0
53	0	0	1	0	0	0	0	1	0	0	0
54	0	0	1	0	0	0	0	1	0	0	0
55	0	0	1	0	0	0	0	0	0	0	0
56	0	0	1	0	0	0	0	0	0	0	0
57	0	0	1	0	0	0	0	0	0	0	0
58	0	0	1	0	0	0	0	0	0	0	0
59	0	0	1	0	0	0	0	0	0	0	0
60	0	0	1	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0

表 15　各个港口每天需准备船只累计数量

天数	港口 A	港口 B	港口 C	港口 D	港口 E	港口 F	港口 G	港口 H	港口 I	港口 J	港口 K
0	2	2	4	4	3	4	3	3	2	1	4
1	4	4	8	8	6	8	6	6	4	2	8
2	6	6	12	12	9	12	9	9	6	3	12
3	8	8	16	16	12	16	12	12	8	4	16
4	10	10	20	20	15	20	15	15	10	5	20
5	12	12	24	24	18	24	18	18	12	6	24

6	14	14	28	28	21	28	21	21	14	7	28
7	16	16	32	32	24	32	24	24	16	8	32
8	18	18	35	36	27	36	27	27	18	9	36
9	20	20	38	40	30	40	30	30	20	10	40
10	22	22	41	44	33	44	33	33	22	11	44
11	24	24	44	48	36	48	36	36	24	12	48
12	26	26	47	52	39	52	39	39	26	13	52
13	28	28	50	56	42	56	42	42	28	13	56
14	30	30	53	60	45	60	44	45	30	13	60
15	32	32	56	64	48	64	46	48	32	13	64
16	34	34	59	68	51	68	48	51	32	13	68
17	36	36	62	72	52	72	49	54	32	13	70
18	38	38	65	76	53	76	50	57	32	13	72
19	40	40	68	80	54	80	51	60	32	13	74
20	42	42	71	84	55	84	52	63	32	13	76
21	44	44	74	88	56	88	53	66	32	13	78
22	46	46	77	92	57	92	54	69	32	13	78
23	48	48	80	96	58	96	55	72	32	13	78
24	50	50	83	100	59	100	56	75	32	13	78
25	52	52	86	104	60	104	57	78	32	13	78
26	54	54	89	108	61	108	57	81	32	13	78
27	56	56	92	112	62	112	57	84	32	13	78
28	58	58	95	113	63	116	57	87	32	13	78
29	60	59	96	114	64	118	57	90	32	13	78
30	62	60	97	115	65	120	57	93	32	13	78

31	64	61	98	116	66	122	57	96	32	13	78
32	66	62	99	117	67	124	57	99	32	13	78
33	68	63	100	117	68	126	57	102	32	13	78
34	70	64	101	117	69	128	57	105	32	13	78
35	72	65	102	117	70	130	57	108	32	13	78
36	74	66	103	117	70	132	57	111	32	13	78
37	76	66	104	117	70	134	57	114	32	13	78
38	78	66	105	117	70	134	57	117	32	13	78
39	80	66	106	117	70	134	57	120	32	13	78
40	82	66	107	117	70	134	57	123	32	13	78
41	84	66	108	117	70	134	57	126	32	13	78
42	86	66	109	117	70	134	57	129	32	13	78
43	87	66	110	117	70	134	57	132	32	13	78
44	88	66	111	117	70	134	57	135	32	13	78
45	88	66	112	117	70	134	57	136	32	13	78
46	88	66	113	117	70	134	57	137	32	13	78
47	88	66	114	117	70	134	57	138	32	13	78
48	88	66	115	117	70	134	57	139	32	13	78
49	88	66	116	117	70	134	57	140	32	13	78
50	88	66	117	117	70	134	57	141	32	13	78
51	88	66	118	117	70	134	57	142	32	13	78
52	88	66	119	117	70	134	57	143	32	13	78
53	88	66	120	117	70	134	57	144	32	13	78
54	88	66	121	117	70	134	57	145	32	13	78
55	88	66	122	117	70	134	57	145	32	13	78

56	88	66	123	117	70	134	57	145	32	13	78
57	88	66	124	117	70	134	57	145	32	13	78
58	88	66	125	117	70	134	57	145	32	13	78
59	88	66	126	117	70	134	57	145	32	13	78
60	88	66	127	117	70	134	57	145	32	13	78
61	88	66	127	117	70	134	57	145	32	13	78

A.4 问题（4）的附录

LINGO 求解调整方案的代码

优化结果.lg4

```
model:

sets:
node/1..11/:port;
line/1..20/:x,c;

link(node,node): M,K;

lin(node, line): E;

endsets

data:
c=
231
120
126
120
135
234
126
135
174
105
141
135
54
246
117
135
294
276
195
108
;

M=
```

```
0 44 108 44 48 116 104 168 88 144 136
44 0 72 12 16 76 64 128 52 112 100
108 72 0 72 64 36 24 68 60 44 52
44 12 72 0 12 76 64 128 60 108 100
48 16 64 12 0 68 60 124 56 104 96
116 76 36 76 68 0 28 68 80 44 68
104 64 24 64 60 28 0 68 46 46 48
168 128 68 128 124 68 68 0 100 52 44
88 52 60 60 56 80 56 100 0 104 60
144 112 44 108 104 44 56 52 104 0 72
136 100 52 100 96 68 48 44 60 72 0;


E=
-1 0 0 0 0      0 0 0 0 0      0 0 0 0 0      0 0 1 1 0
0 -1 0 1 0      0 0 0 0 0      0 0 0 1 0      0 0 0 0 0
0 1 -1 -1 0     0 0 0 0 0      0 0 1 0 0      0 0 0 0 0
0 0 1 0 -1     -1 0 0 0 0      0 0 0 0 0      0 0 0 1 0
0 0 0 0 0       0 -1 -1 0 0    0 1 0 0 0      0 0 0 0 0
1 0 0 0 1       0 0 0 -1 -1   -1 0 0 0 0      0 0 0 0 0
0 0 0 0 0       0 0 1 0 0      0 -1 -1 0 1    0 0 0 0 1
0 0 0 0 0       1 0 0 0 0      0 0 0 -1 -1    0 0 0 0 0
0 0 0 0 0       0 1 0 1 0      0 0 0 0 0     -1 0 0 0 0
0 0 0 0 0       0 0 0 0 1      0 0 0 0 0      0 -1 0 0 0
0 0 0 0 0       0 0 0 0 0      1 0 0 0 0      1 0 -1 -1 -1;


enddata

max=@sum(line(i): c(i)*x(i)) - @sum(node(i): @sum(node(j): M(i,j)*@abs(K(i,j)) ));


@for(node(j): (@sum(line(i): x(i)*E(j,i) ) - @sum(node(i): K(j,i)-K(i,j) ) ) = 0 );

@sum(line(i): x(i)) <= 32;

@for(line(i): x(i) <=3);

@for(line(i): x(i) >=0);

!调度次数;
```

```
! @sum(node(i): @sum(node(j): @abs(K(i,j)) )) ;

@for(node(i): @for(node(j): @gin(K(i,j))));


!@for(node(i): @for(node(j): @free(K(i,j))));


end
```