

# 《运筹学》课程设计 ——指导书

吴薇薇 陈新元 编著

## 目录

1. 教学目的.....	3
2. 教学计划.....	3
3. 课程设计任务 .....	3
4. 基于连接网络的机型指派 .....	3
4.1. 符号与参数说明 .....	5
4.2. 连接网络法模型的优化目标和约束条件 .....	6
4.2.1. 目标函数 .....	7
4.2.2. 航班覆盖约束 .....	8
4.2.3. 流平衡约束 .....	9
4.2.4. 机队规模约束 .....	10
4.2.5. 完整数学模型 .....	10
5. 基于时空网络的机型指派 .....	11
5.1. 时空网络优化模型的参数定义 .....	15
5.2. 时空网络优化模型的目标函数与约束条件 .....	16
6. 实例数据 .....	20
7. 课设任务 .....	24
8. 课程设计报告格式要求 .....	24
附件 .....	1
附件 1 Gurobi 安装与使用方法 .....	1
附件 2 Lindo/Lingo 功能与使用方法 .....	2
附件 3 时空网络 .....	20
附件 4 设计报告模板 .....	23

## 1. 教学目的

《运筹学》课程设计是继“运筹学”课之后开出的实践环节课程。其目的是训练学生综合运用学过的运筹学基本知识，独立解决民航运输管理决策问题和提出决策建议的能力。

## 2. 教学计划

《运筹学》课程设计时间要求二至三周，以 4-8 小时/天计算，要求学生投入 56 小时的时间，安排在课程教学结束之后进行。

## 3. 课程设计任务

培养学生利用所学的运筹学知识，使用商业求解器（Gurobi、Lingo），进行（1）基于连接网络的机型指派，或者（2）基于时空网络的机型指派（二选一）的优化，训练学生应用运筹学和优化工具，独立解决航班计划决策问题和提出决策建议的能力。

### （1）机型指派问题

机型指派优化设计是航空公司制订航班计划的重要内容，它要求在满足航班频率和时刻安排以及各机型飞机总数约束的条件下，将各机型飞机指派给相应的航班，使运行成本最小化。本课程设计要求建立机型指派问题的数学模型，应用优化软件（Gurobi、Lingo）进行建模求解，给出决策建议，包括各机型执行的航班子集和相应的运行成本。

### （2）问题描述

机型分配问题是航空运输领域中的一个重要问题，其目标是将不同机型的飞机分配到不同的航线上，以最大化航空公司的利润。为了解决这个问题，通常需要在航线网络的基础上构造 0-1 整数规划数学模型。在构建航线网络时，主要有两种优化方法：连接网络法和时空网络法。

## 4. 基于连接网络的机型指派

连接网络法是一种基于航班之间的连接关系建立模型的方法，连接网络法首先是由 Abara 于 1989 年提出，连接网络图主要由节点、弧线和时间线构成。节点包含两类：一类节点表示航班的始发和到达时间节点，另一类节点为构造的节点用来表示起始弧和终止弧。其中，连接网络法的弧线共有三种：连接弧、航班弧和起始/终止弧。时间线则按照航班运行时间分为到达时间线和离开时间线。

基于连接网络法的机型指派模型建立的核心思想是为每个机队分配一个子网络，以使该机队的飞机可以顺利地执行子网络中的航班任务。每个机队子网络都会受到一组覆盖约束的限制，以确保每个航班都能被分配到一个机队子网络中。通过这种方式，机队之间的相互影响被最小化，从而实现更加有效的航班规划和调度。对于连接网络图而言，假定每天的实际航班计划大致相同。如图 1 所示，针对机场 A、机场 B，机型指派问题是指针对每天的航班计划进行优化，因此本文用第 N 天的航班计划来表示连接网络图。

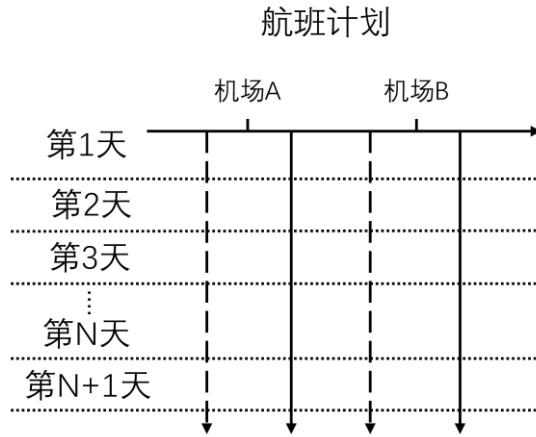


图 1 航班计划

我们假设第  $N$  天的航班计划仅涉及机场 A 和机场 B 之间的航班调度。同时，为了维持网络流量的平衡和确保第二天航班计划的正常执行，机队  $K$  中的所有机型在当天晚上返回原机场的数量和当天早上从机场出发的数量应该相同。基于以上假设，我们建立了第  $N$  天的航班计划连接网络图，具体如图 2 所示。

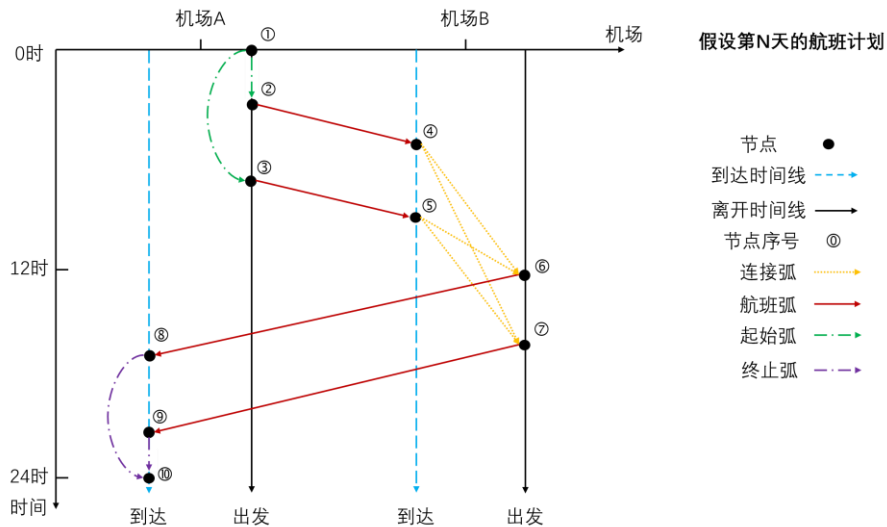


图 2 第  $N$  天的航班计划连接网络图

在连接网络图中，纵轴为时间轴，横轴表示机场。纵轴的时间按一天 24 小时制划分，起始弧上的节点①对应 0 时，终止弧上的节点⑩对应 24 时。横轴共有两个机场为机场 A 和机场 B，其机场内的航班调度分为到达时间线和出发时间线，分别用不同颜色的虚线和实线表示。节点则代表航班的出发和到达的时间点，到达时间线上的节点表示不同航班不同时间到达该机场并停留，出发时间线上的节点表示航班由该时间从此机场出发。

连接网络法中有三种弧线。航班弧代表机场之间的不同航班；连接弧表示一个到达航班和一个出发航班之间可能的飞机连接；起始弧代表前一天停留在原机场并在一天开始时从此机场出发的飞机，终止弧代表一天中到达和最终停留在此机场的飞机。

如图 2 所示，其中①→②为起始弧 $S_1$ ，①→③为起始弧 $S_2$ 。⑧→⑩为终止弧 $f_1$ ，⑨→⑩为

终止弧 $f_2$ 。起始弧 $S_1$ 表示第  $N-1$  天停留机场 A 的机队 $K$ 在机型分配后出发并离开机场 A，终止弧 $f_1$ 则表示第  $N$  天最后的机型指派，机队 $K$ 停留在机场 A 过夜，再于第  $N+1$  天从机场 A 出发。在图中，我们使用不同颜色的虚实线来区分这三种不同类型的弧线。图中航班弧用红色实线表示，如②→④、③→⑤皆为从机场 A 出发抵达机场 B 的航班。连接弧用黄色虚线表示，如④→⑥、④→⑦都为机场 B 中到达航班和出发航班可能的飞机连接。由于连接弧在连接网络法中代表的是可能的飞机连接，因此对于节点④而言，时间顺序之后的节点⑥和节点⑦都可以作为机场 B 调度飞机出发的时间节点。为了保证每种分配方案的可能性，航班之间所有可行的连接都必须指定，在本图中，机场 B 共有 4 种连接方案。然而，这种指定所有可行航班连接的方法可能会导致航班之间的连接数增多，也是连接网络法复杂性的一个体现。

#### 4.1. 符号与参数说明

根据上一小节对第  $N$  天连接网络图的假设，我们可以将其简化为第  $N$  天的航班计划拓扑图，以便更好地定义数学模型中的变量并解释最终的连接网络模型。简化后的图如下图 3 所示。

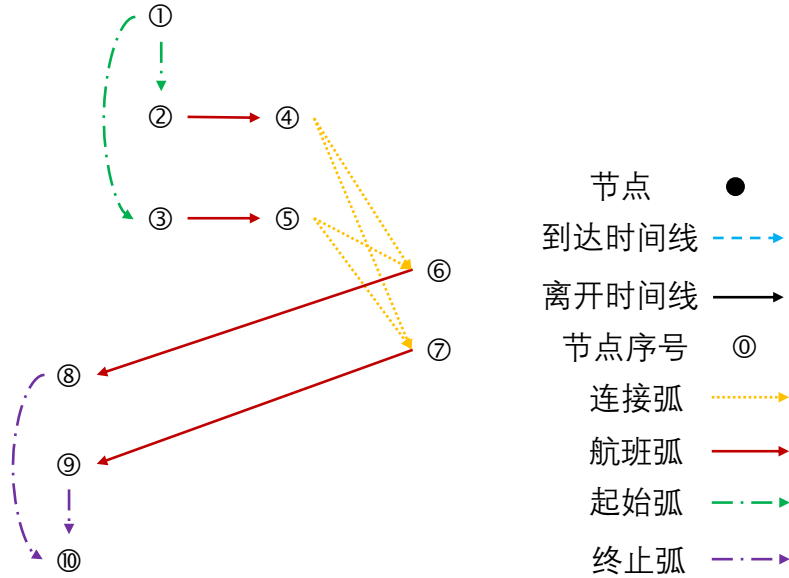


图 3 第  $N$  天的航班计划拓扑图

数学模型中的参数包含四种类型，分别为集合、常量、参量以及决策变量，具体符号及解释由表格 1 给出。

表格 1 连接网络法模型符号与参数表

集合	
$K = \{k_1, k_2\}$	机队 $K$ 由两种机型 $k_1, k_2$ 组成
$L = \{i, j, l\}$	$L$ 表示航班集合，由 $i, j, l$ 三类航班组成；其中 $i$ 表示航班 $l$ 的前序航班，航班 $j$ 表示 $l$ 的后续航班
$L^+ = L \cup \{0\}$ $= \{i, j, l, 0\}$	$L^+$ 表示航班集合包含起始弧和终止弧的情况，为了保证网络流平衡和到达出发飞机的数量守恒，应在航班集合中假设 0 元素代表起/终弧。其中 $i = 0$ 表示起始弧， $j = 0$ 表示终止弧
$S = \{a, b\}$	集合包含机场 A 和机场 B

$L_S^A$	到达机场 $S$ 的航班集合
$L_S^D$	从机场 $S$ 出发的航班集合
<b>常量</b>	
$M_k = 2$	机队 $K$ 的可用总飞机数量为 2
<b>参量</b>	
$c_k$	机队 $K$ 中每架飞机的成本
$p_{jk}$	运营由机型 $k$ 执行的航班 $j$ 的收益
<b>决策变量</b>	
$x_{ijk} \in \{0,1\}$	当机队 $k$ 覆盖 $i \rightarrow j$ 的连接时, $x_{ijk} = 1$ ; 否则, $x_{ijk} = 0$

针对上述四种参数,结合航班拓扑图作出更具体的解释。对于航班集合  $L = \{i, j, l\}$  而言,在假定的第  $N$  天航班计划中,其可以表示为  $L = \{i, j, l\} = \{(2,4), (3,5), (6,8), (7,9)\}$ , 其中如上图例(2,4)代表②→④的航班弧。

因此,  $L^+ = L \cup \{0\} = \{(2,4), (3,5), (6,8), (7,9), 0\}$ , 当  $i = 0 = (1,2), (1,3)$  表示起始弧,  $j = 0 = (8,10), (9,10)$  表示终止弧; 所以, 对于一个航班连接  $i \rightarrow j$ ,  $i, j \in L^+$ , 如果  $i = 0$ , 那么  $j$  是每天飞机航线的第一条航班; 如果  $j = 0$ , 那么  $i$  是每天飞机航线的最后一条航班。在图 3 中, 连接弧集合则为  $\{(4,6), (4,7), (5,6), (5,7)\}$ , 表示 4 机场 B 内全部可行的航班连接。

#### 4.2. 连接网络法模型的优化目标和约束条件

简化的航班计划拓扑图是一种简洁且易于理解的表示方法,可以用于解释参数设定。然而,为了使我们基于连接网络法的机型指派模型更加直观,我们还可以对航班计划拓扑图进行简单的网络变换。具体而言,我们可以使用节点序号来代表连接网络法中的三种弧线: 航班弧、连接弧、起始/终止弧。同时,原本在连接网络图中用来代表弧线的箭头,可以用来表示航班之间的连接,这样在判断前序和后序航班时更为方便。通过这种方式,我们可以得到简洁明了的连接网络法模型,其网络变换后的航班计划拓扑图如图 4 所示。

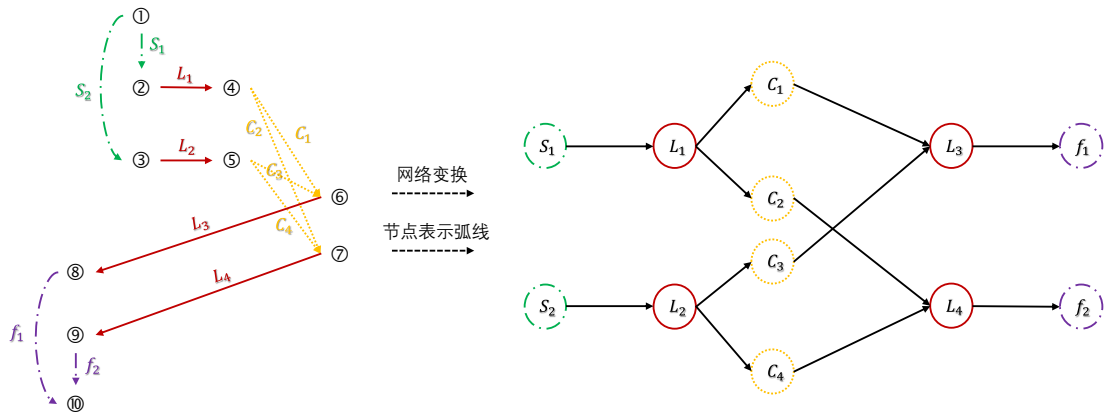


图 4 网络变换后的航班计划拓扑图

在图 4 中, 起始弧为  $S_1, S_2$ , 终止弧为  $f_1, f_2$ , 在上一小节已经解释过。航班集合  $L =$

$\{i,j,l\} = \{(2,4),(3,5),(6,8),(7,9)\}$ ，在变换后的拓扑图中，分别用 $L_1$ 、 $L_2$ 、 $L_3$ 、 $L_4$ 表示。连接弧 $\{(4,6),(4,7),(5,6),(5,7)\}$ 则用 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ 来代替。

但在实际建立模型的过程中，连接弧如 $C_1$ 可以用航班连接 $L_1 \rightarrow L_3$ 来代替，因此对于决策变量 $x_{L_1 L_3 k}$ ，当机型 $k$ 覆盖 $L_1 \rightarrow L_3$ 的连接时， $x_{L_1 L_3 k} = 1$ ；否则， $x_{L_1 L_3 k} = 0$ 。本文为了方便理解，将图 4 的网络变换后的拓扑图作简化，删去连接弧，依旧用节点代表弧并在此基础上表示模型，简化后的图如下图 5 所示。

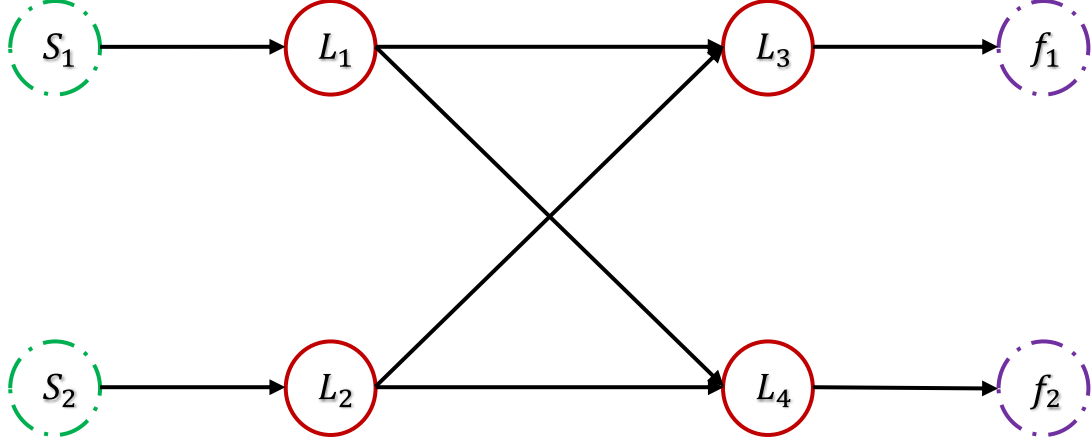


图 5 简化后的航班计划拓扑图

#### 4.2.1. 目标函数

机型指派问题的最终目标是最大化航空公司的经营利润，因此连接网络模型的目标函数为求利润的最大值，而航空公司利润可以简化等于收益减成本，所以

目标函数 =  $Max$ （当天所有航段的收益 - 当天调度所使用的飞机成本）

对于图 5 而言，可得目标函数为  $Max[(p_{L_1 k_1} x_{S_1 L_1 k_1} + p_{L_1 k_2} x_{S_1 L_1 k_2} + p_{L_2 k_1} x_{S_2 L_2 k_1} + p_{L_2 k_2} x_{S_2 L_2 k_2} + p_{L_3 k_1} x_{L_1 L_3 k_1} + p_{L_3 k_2} x_{L_1 L_3 k_2} + p_{L_3 k_1} x_{L_2 L_3 k_1} + p_{L_3 k_2} x_{L_2 L_3 k_2} + p_{L_4 k_1} x_{L_2 L_4 k_1} + p_{L_4 k_2} x_{L_2 L_4 k_2} + p_{L_4 k_1} x_{L_1 L_4 k_1} + p_{L_4 k_2} x_{L_1 L_4 k_2}) - (c_{k_1} x_{S_1 L_1 k_1} + c_{k_2} x_{S_1 L_1 k_2} + c_{k_1} x_{S_2 L_2 k_1} + c_{k_2} x_{S_2 L_2 k_2})]$ 。

对于模型中涉及的航段，上述式子我们还可以用表格的形式列出，便于读者理解，如下表格 2 所示。表格的左侧表示针对某个航段，机型 $k_1$ 、 $k_2$ 的不同收益公式，将左侧累加后用中间的符号与右侧累加后的公式相连接；表格右侧表示针对不同航段，不同机型的成本公式。

表格 2 目标函数

航段连接	左侧（收益）		符号	右侧（成本）	
	机型 $k_1$	机型 $k_2$		机型 $k_1$	机型 $k_2$
$S_1 \rightarrow L_1$ (起始)	$p_{L_1 k_1} x_{S_1 L_1 k_1}$	$p_{L_1 k_2} x_{S_1 L_1 k_2}$	—	$c_{k_1} x_{S_1 L_1 k_1}$	$c_{k_2} x_{S_1 L_1 k_2}$

$S_2 \rightarrow L_2$ (起始)	$p_{L_2 k_1} x_{S_2 L_2 k_1}$	$p_{L_2 k_2} x_{S_2 L_2 k_2}$	—	$c_{k_1} x_{S_2 L_2 k_1}$	$c_{k_2} x_{S_2 L_2 k_2}$
$L_1 \rightarrow L_3$	$p_{L_3 k_1} x_{L_1 L_3 k_1}$	$p_{L_3 k_2} x_{L_1 L_3 k_2}$	—		
$L_1 \rightarrow L_4$	$p_{L_4 k_1} x_{L_1 L_4 k_1}$	$p_{L_4 k_2} x_{L_1 L_4 k_2}$	—		
$L_2 \rightarrow L_3$	$p_{L_3 k_1} x_{L_2 L_3 k_1}$	$p_{L_3 k_2} x_{L_2 L_3 k_2}$	—		
$L_2 \rightarrow L_4$	$p_{L_4 k_1} x_{L_2 L_4 k_1}$	$p_{L_4 k_2} x_{L_2 L_4 k_2}$	—		

将上述函数化简可得优化目标函数公式(1)，如下所示：

$$\text{Max} \quad \sum_{i \in L^+} \sum_{j \in L} \sum_{k \in K} p_{jk} x_{ijk} - \sum_{j \in L} \sum_{k \in K} c_k x_{0jk} \quad (1)$$

(1)中的目标函数由两部分组成。第一部分是所有航段获得的利润，第二部分是所用飞机数量的成本。这里  $\sum_{j \in L} x_{0jk}$  表示机队  $K$  当天所使用的总飞机数。

#### 4.2.2. 航班覆盖约束

为了使航班计划中的每一个航班都匹配一种现有机型，航班覆盖约束(2)被提出。其可以表示为对于  $\forall j \in L$ ，都存在  $x_{S_1 L_1 k_1} + x_{S_1 L_1 k_2} = 1$ ,  $x_{S_2 L_2 k_1} + x_{S_2 L_2 k_2} = 1$ ,  $x_{L_1 L_3 k_1} + x_{L_1 L_3 k_2} + x_{L_2 L_3 k_1} + x_{L_2 L_3 k_2} = 1$ ,  $x_{L_1 L_4 k_1} + x_{L_1 L_4 k_2} + x_{L_2 L_4 k_1} + x_{L_2 L_4 k_2} = 1$ 。

上述式子，表格列出后如下表格 3 所示，左侧为每个航班的决策变量，右侧判断航段是否被一种机型覆盖，左侧连接为加号。

表格 3 航班覆盖约束

航班	航段连接	左侧（每个航班）		符号	右侧（判断每个航班是否都匹配一种机型）
		机型 $k_1$	机型 $k_2$		
$L_1$	$S_1 \rightarrow L_1$ (起始)	$x_{S_1 L_1 k_1}$	$x_{S_1 L_1 k_2}$	=	1
$L_2$	$S_2 \rightarrow L_2$ (起始)	$x_{S_2 L_2 k_1}$	$x_{S_2 L_2 k_2}$	=	1
$L_3$	$L_1 \rightarrow L_3$ 或 $L_2 \rightarrow L_3$	$x_{L_1 L_3 k_1}$ + $x_{L_2 L_3 k_1}$	$x_{L_1 L_3 k_2}$ + $x_{L_2 L_3 k_2}$	=	1
$L_4$	$L_1 \rightarrow L_4$ 或 $L_2 \rightarrow L_4$	$x_{L_1 L_4 k_1}$ + $x_{L_2 L_4 k_1}$	$x_{L_1 L_4 k_2}$ + $x_{L_2 L_4 k_2}$	=	1

简化后可得约束条件(2)，公式如下：



$$\sum_{i \in L^+} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in L \quad (2)$$

#### 4.2.3. 流平衡约束

设备连续性约束(3)要求在同一航线段（Flight Leg）开始或结束的两个连接应由同一机型承载。简而言之，对于具有相同航班号的一次经停航班的两个航段，必须分配相同的机型。因此，网络中的流量平衡得以维持。可以表示为对于  $\forall l \in L, \forall k \in K$ ，都存在  $x_{S_1 L_1 k_1} - (x_{L_1 L_3 k_1} + x_{L_1 L_4 k_1}) = 0$ ， $x_{S_2 L_2 k_1} - (x_{L_2 L_3 k_1} + x_{L_2 L_4 k_1}) = 0$ ， $(x_{L_1 L_3 k_1} + x_{L_2 L_3 k_1}) - x_{L_3 f_1 k_1} = 0$ ， $(x_{L_1 L_4 k_1} + x_{L_2 L_4 k_1}) - x_{L_4 f_2 k_1} = 0$ 。

同理得机型  $k_2$  的约束， $x_{S_1 L_1 k_2} - (x_{L_1 L_3 k_2} + x_{L_1 L_4 k_2}) = 0$ ， $x_{S_2 L_2 k_2} - (x_{L_2 L_3 k_2} + x_{L_2 L_4 k_2}) = 0$ ， $(x_{L_1 L_3 k_2} + x_{L_2 L_3 k_2}) - x_{L_3 f_1 k_2} = 0$ ， $(x_{L_1 L_4 k_2} + x_{L_2 L_4 k_2}) - x_{L_4 f_2 k_2} = 0$ 。

上述式子，表格列出后如下表格 4 所示，左侧累加表示前序航段，右侧表示后续航段，符号等号代表两个航段连接应由同一机型承载。

表格 4 连续性约束

航班	航段连接	机型	左侧（前序航段）	符号	航段连接	右侧（后续航段）
$L_1$	$S_1 \rightarrow L_1$ (起始)	机型 $k_1$	$x_{S_1 L_1 k_1}$	$=$	$L_1 \rightarrow L_3$ 或 $L_1 \rightarrow L_4$	$x_{L_1 L_3 k_1} + x_{L_1 L_3 k_2}$
		机型 $k_2$	$x_{S_1 L_1 k_2}$			$x_{L_1 L_4 k_1} + x_{L_1 L_4 k_2}$
$L_2$	$S_2 \rightarrow L_2$ (起始)	机型 $k_1$	$x_{S_2 L_2 k_1}$	$=$	$L_2 \rightarrow L_3$ 或 $L_2 \rightarrow L_4$	$x_{L_2 L_3 k_1} + x_{L_2 L_4 k_1}$
		机型 $k_2$	$x_{S_2 L_2 k_2}$			$x_{L_2 L_3 k_2} + x_{L_2 L_4 k_2}$
$L_3$	$L_1 \rightarrow L_3$ 或 $L_2 \rightarrow L_3$	机型 $k_1$	$x_{L_1 L_3 k_1} + x_{L_2 L_3 k_1}$	$=$	$L_3 \rightarrow f_1$ (终止)	$x_{L_3 f_1 k_1}$
		机型 $k_2$	$x_{L_1 L_3 k_2} + x_{L_2 L_3 k_2}$	$=$		$x_{L_3 f_1 k_2}$
$L_4$	$L_1 \rightarrow L_4$ 或 $L_2 \rightarrow L_4$	机型 $k_1$	$x_{L_1 L_4 k_1} + x_{L_2 L_4 k_1}$	$=$	$L_4 \rightarrow f_2$ (终止)	$x_{L_4 f_2 k_1}$
		机型 $k_2$	$x_{L_1 L_4 k_2} + x_{L_2 L_4 k_2}$	$=$		$x_{L_4 f_2 k_2}$

将上述公式合并可写为设备连续性约束(3)，如下：

$$\sum_{i \in L^+} x_{ilk} - \sum_{j \in L^+} x_{ljk} = 0 \quad \forall l \in L, \forall k \in K \quad (3)$$

约束条件(4)表示航班调度平衡约束，其作用是确保每个机场在一天开始时起飞的飞机数量等于在同一天结束时降落的飞机数量，从而保证航班计划可以每天重复执行。例如对于机场 A，机队  $K$  中的所有机型都必须在当天晚上返回原机场。对于  $\forall s \in S, \forall k \in K$ （本文仅包含两个机场），都存在  $(x_{S_1 L_1 k_1} + x_{S_2 L_2 k_1}) - (x_{L_3 f_1 k_1} + x_{L_4 f_2 k_1}) = 0$ ， $(x_{S_1 L_1 k_2} + x_{S_2 L_2 k_2}) - (x_{L_3 f_1 k_2} + x_{L_4 f_2 k_2}) = 0$ 。

表格列出后如下表格 5 所示，左侧表示机场 A 的出发飞机数量，右侧表示机场 B 的到达飞机数量，符号等号要求数量相等。

表格 5 航班调度平衡约束

左侧（出发航班）			符号	右侧（到达航班）		
航段连接	机型 $k_1$	机型 $k_2$		航段连接	机型 $k_1$	机型 $k_2$
$S_1 \rightarrow L_1$ (起始)	$x_{S_1 L_1 k_1}$	$x_{S_1 L_1 k_2}$	=	$L_3 \rightarrow f_1$ (终止)	$x_{L_3 f_1 k_1}$	$x_{L_3 f_1 k_2}$
$S_2 \rightarrow L_2$ (起始)	$x_{S_2 L_2 k_1}$	$x_{S_2 L_2 k_2}$	=	$L_4 \rightarrow f_2$ (终止)	$x_{L_4 f_2 k_1}$	$x_{L_4 f_2 k_2}$

简化后可得公式(4):

$$\sum_{l \in L_S^D} x_{olk} - \sum_{l \in L_S^A} x_{lok} = 0 \quad \forall s \in S, \forall k \in K \quad (4)$$

(5)是飞机数量约束条件，旨在确保所调度和分配的飞机数量不会超过每个机队可用飞机的总数。表示为对于 $\forall k \in K$ ，都存在 $x_{S_1 L_1 k_1} + x_{S_2 L_2 k_1} \leq M_k$ ， $x_{S_1 L_1 k_2} + x_{S_2 L_2 k_2} \leq M_k$ 。

#### 4.2.4. 机队规模约束

上述式子，表格列出后如表格 6 所示，左侧表示总调度的飞机数量，右侧机队 K 总共可用的飞机数量，符号小于等于要求不超过机队可用飞机数。

表格 6 飞机数量约束平衡

航段连接	左侧（调度飞机数）		符号	右侧（机队可用飞机数）
	机型 $k_1$	机型 $k_2$		
$S_1 \rightarrow L_1$ (起始)	$x_{S_1 L_1 k_1}$	$x_{S_1 L_1 k_2}$	$\leq$	$M_k$
$S_2 \rightarrow L_2$ (起始)	$x_{S_2 L_2 k_1}$	$x_{S_2 L_2 k_2}$	$\leq$	$M_k$

简化后可得公式(5)如下:

$$\sum_{l \in L} x_{olk} \leq M_k \quad \forall k \in K \quad (5)$$

式(6)表示 $x_{ijk}$ 为决策变量。

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in L^+, \forall k \in K \quad (6)$$

#### 4.2.5. 完整数学模型

综上所述，基于连接网络法的机型指派模型建立完成，该模型的优化目标函数和约束条件如下所示:

$$\text{Max} \quad \sum_{i \in L^+} \sum_{j \in L} \sum_{k \in K} p_{jk} x_{ijk} - \sum_{j \in L} \sum_{k \in K} c_k x_{0jk} \quad (7)$$

s. t.

$$\sum_{i \in L^+} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in L \quad (8)$$

$$\sum_{i \in L^+} x_{ilk} - \sum_{j \in L^+} x_{ljk} = 0 \quad \forall l \in L, \forall k \in K \quad (9)$$

$$\sum_{l \in L_S^D} x_{0lk} - \sum_{l \in L_S^A} x_{l0k} = 0 \quad \forall s \in S, \forall k \in K \quad (10)$$

$$\sum_{l \in L} x_{0lk} \leq M_k \quad \forall k \in K \quad (11)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in L^+, \forall k \in K \quad (12)$$

## 5. 基于时空网络的机型指派

在 Hane 等人提出的时空网络图中，网络由节点和弧构成，横轴表示机场类型，纵轴表示时间，如图 6 所示。但与连接网络图不同的是，时空网络图共有三种弧线分别为航班弧、地面弧与环绕弧。其中，时空网络中的弧是指连接两个节点的有向弧。每个航班对应于一个从机场出发（事件开始）和到达机场（事件结束）的航段弧。而每个地面弧连接同一机场的两个连续节点。环绕弧则是地面弧的一种特殊类型，它连接某个机场的最后一个节点和第一个节点。环绕弧表示的是飞机在连续两天之间的循环，该机型在机场过夜。作为最后一个进港航班，它将在下一次航班计划中与第二天的出港航班衔接起来，过夜期间可以对飞机进行检修及保养等工作。三种弧线的具体解释如表格 7 所示。而与连接网络法不同的是，连接网络法用到达时间线和出发时间线区分事件，节点仅表示时间与位置。时空网络法的节点则范围更广，作为活动节点包含特定事件。

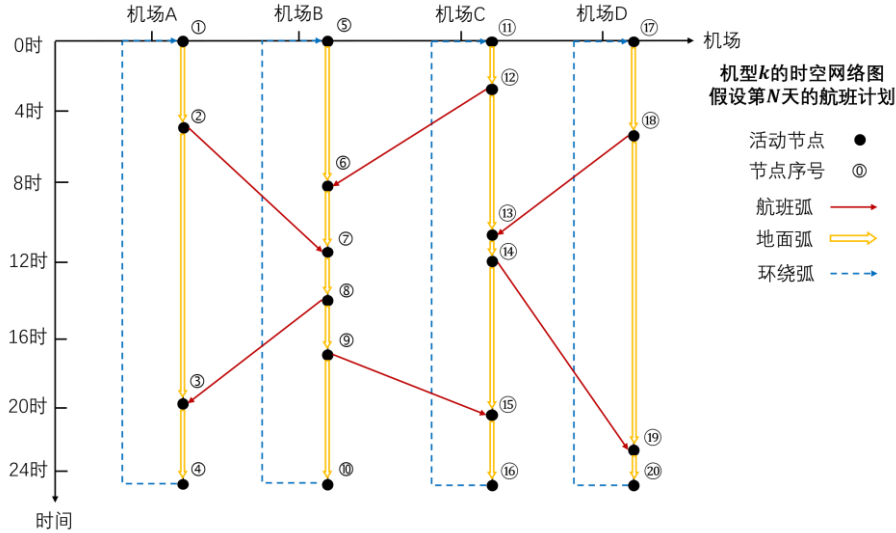


图 6 单一机型的时空网络图

表格 7 时空连接图的三种弧线

弧	含义
航班弧	航班弧 (Leg Arc) 连接不同机场之间的航班, 航段弧代表飞机从一个机场起飞, 飞行结束到达另一个机场的航程。
地面弧	地面弧 (Ground Arc) 表示停留在机场地面上的飞机, 进行乘客换乘、货物装卸、燃料补给等操作。
环绕弧	环绕弧 (Wrap-around Arc) 连接一个机场的最后一个节点和该机场的第一个节点。它表示飞机在该机场过夜, 并准备进行下一次航班计划的航班。

网络图由一系列的活动节点构成, 每个节点都代表一个特定的事件。这些事件主要分为两类: 出发事件和到达事件。这两种事件都与相应的活动节点有一一对应的关系, 而节点的时间和位置则直接反映了其对应事件的具体发生时间和地点。

需要注意的是, 当两个或多个事件的时间和地点完全一致时, 这些事件在网络图中只会被表示为一个单一的节点。根据这个规则, 我们可以理解, 每个航班实际上对应的是一个出发事件和一个到达事件。整个网络图的构建, 都是基于这些事件集合来完成的。

因此, 在此基础上, 我们需要明确两个约束条件的定义。

1.最小连接时间: 在网络中我们需要考虑航班轮换时两个连续航班之间的最小机场连接时间, 在航班轮换中, 地面上的飞机需要维修、加油等操作时间、旅客的下机时间或者货物的登机时间等。每个到达事件在网络中对应一个活动节点, 到达事件的发生时间表示为航班的实际到达时间和最小机场连接时间的总和。在这样的设置下, 如果一个到达事件连接到另一个出发事件, 到达和出发之间的实际时间间隔必须大于最小连接时间。

2.节点流量守恒: 为了保持网络中的流量平衡, 我们需要确保每个活动节点上的流量平衡。每个活动节点都有输入流量和输出流量, 我们需要保证节点流量平衡, 如下图 7 表示。

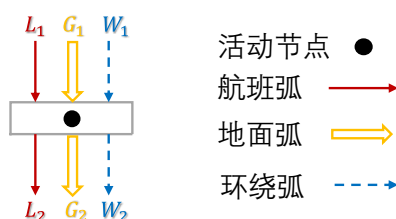


图 7 节点流量守恒

对于该活动节点, 分别有一个输入航班弧 $L_1$ 、一个输入地面弧 $G_1$ 和一个输入环绕弧 $W_1$ ; 一个输出航班弧 $L_2$ 、一个输出地面弧 $G_2$ 和一个输出环绕弧 $W_2$ 。我们用表示符号代替其流量, 例如 $L_1$ 代表输入航班弧的流量, 因此流量守恒可以表示为:

$$L_1 + G_1 + W_1 = L_2 + G_2 + W_2$$

值得注意的是, 活动节点可能存在多个输入航班弧和输出航班弧, 而所有的输入航班弧和输出航班弧都应考虑在流量平衡约束中。除此之外, 环绕弧对于每个节点也并不是都存在, 通常情况下只有起始节点和末端节点存在环绕弧, 表示前一天过夜的飞机和准备在该机场过

夜的飞机。

综合上述解释，我们可以在单一机型条件下为机型 $k$ 构建其时空网络图，因为不考虑相同类型的飞机可以互换使用，每个机型都有其单独的时空网络图。我们先对单一机型 $k$ 和假定第 $N$ 天航班计划作出下述时空网络图，如图 8 所示。

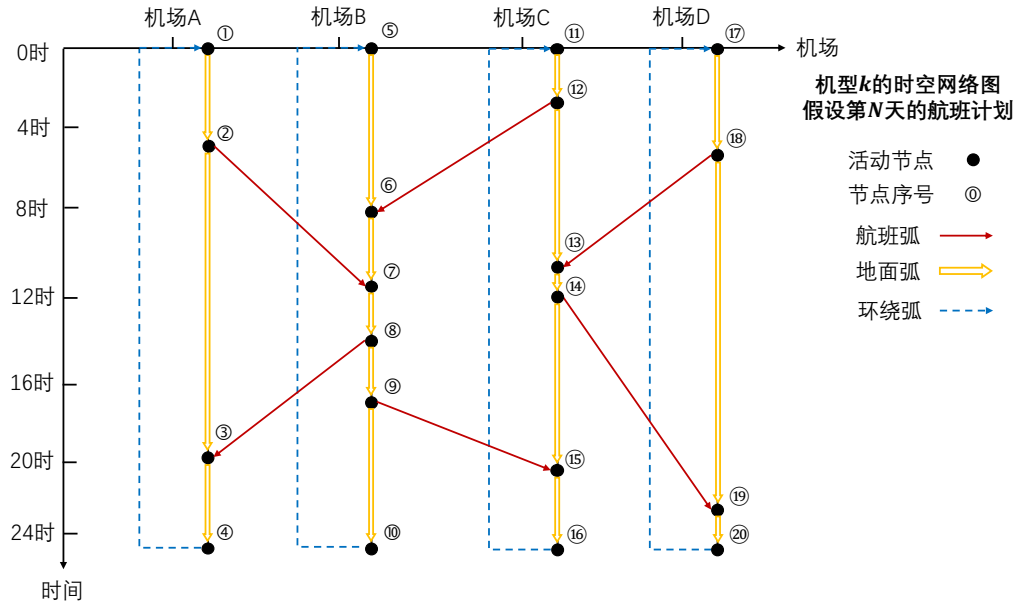


图 8 单一机型的时空网络图

在图 8 中，横轴表示机场，纵轴表示时间，事件的到达或出发均由有向箭头即航班弧表示。例如②→⑦表示航班从机场 A 出发，最终到达机场 B 的事件。但在实际航线网络设计中，针对每个机型画出相应时空网络图显然太过复杂，而且单一机型或相同配置飞机可以替代的假设并不通用。因此，我们假定共有 3 种机型，将不同机型合并在一张时空网络图中，为便于理解，我们在图中减少航班数量，如下图 9 所示。

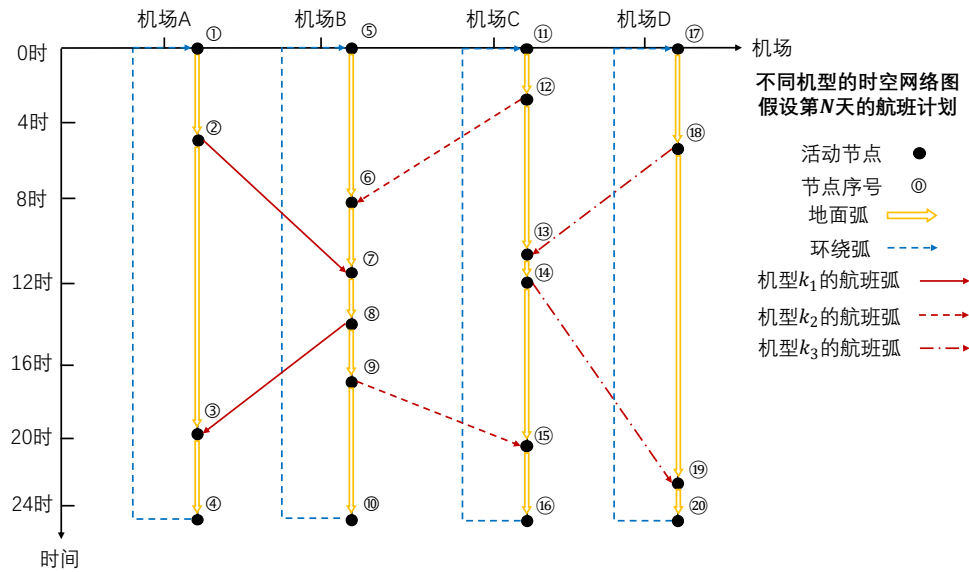


图 9 不同机型的时空网络图

在图中我们可以清晰的看到，横轴共四种不同类型的机场，不同的机型用不同的线条形

式表现出来。相较连接网络图，我们可以明显的看到，在机场数量和机型数量都增加的情况下，时空网络图的决策变量数仍小于连接网络图的决策变量，这说明连接网络法在计算上更为复杂。

在图 9 中，每个机场每个机型早上出发的航班数量等于晚上到达的航班数量，这是为了保证航班计划都能每天重复执行。

②→⑦表示机型 $k_1$ 的航班弧，从机场 A 出发到达机场 B；⑫→⑥表示机型 $k_2$ 的航班弧，由机场 C 出发到达机场 B；⑮→⑬表示机型 $k_3$ 从机场 D 出发到达机场 C。

而地面弧，例如②→③连接两个活动节点，表示还在地面上停留的飞机；环绕弧则连接机场时间线上的第一个节点与末端节点，如图中④→①表示飞机在机场 A 过夜，并准备进行第二天航班任务，因此④其实连接的是第 $N + 1$ 天的①节点。

此外，图中地面弧⑬→⑭表示飞机由机场 D 出发后到达机场 C，经过短暂时间后执行航班，并不为一个活动节点。因为时空网络图节点表示时间 = 航班实际到达时间 + 最小连接时间，航班到达机场时需要维修、加油等操作时间，地面弧⑬→⑭中的到达与出发时间间隔也应大于最小连接时间。

而为了建立模型的目标函数与约束条件，我们应计算出网络中飞机的数量并使其满足各种流量守恒。但由于时空网络图的特性，空中飞机的数量与地面飞机的数量并未分别在图中显示，因此，时空网络法引入一个新的计数方法：计数时间线。该方法的核心思想是对某一时刻的时空网络图进行定格，我们将定格瞬间的时间点定义为时空网络的计数线，以此计算该时刻的空中与地面的飞机数量，如图 10 所示。

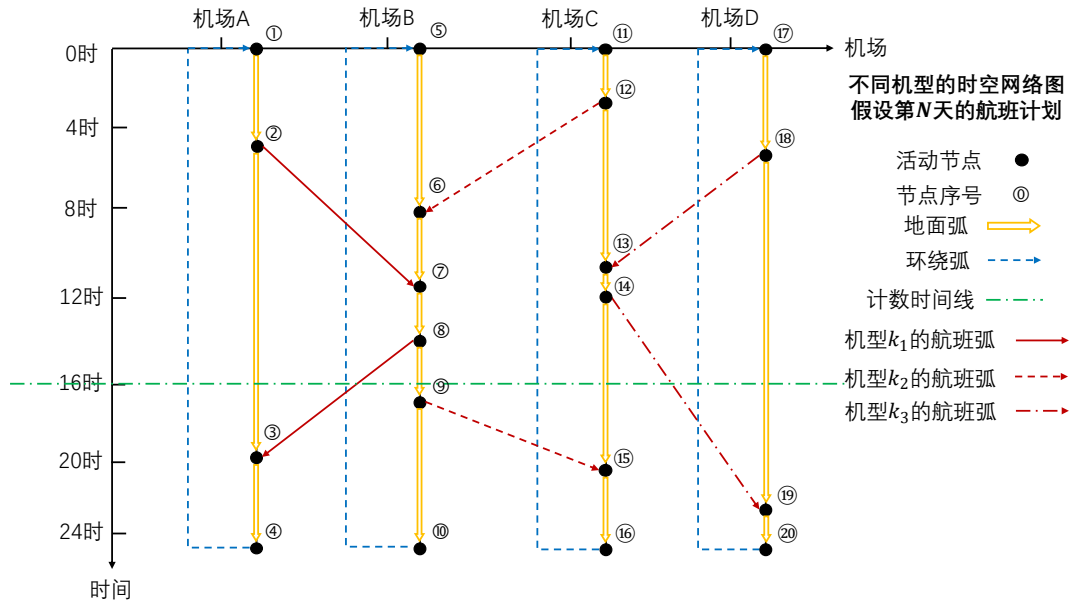


图 10 考虑计数时间线的时空网络图

计数时间线可以任意取时刻，在图 10 中，我们取计数时间线为 16 时。穿过计数线的流量，包括航班弧和地面弧，表示空中与地面的飞机总数。例如⑧→③为机场 B 到机场 A 采用机型 $k_1$ 的航班，其穿过计数时间线，所以在计数时刻将机型 $k_1$ 计入仍在空中的飞机数量。而地面弧⑧→⑨同样在计数时刻穿越计数时间线，所以机型 $k_2$ 被计入仍在地面的飞机数量。

据此，我们需要确保穿过该计数线的流量要小于可用飞机的数量。

### 5.1. 时空网络优化模型的参数定义

同样的，我们采取前几章所用的简化方法，将节点与弧线抽象出来作出时空网络法的航班拓扑图，如下图所示 11 所示。

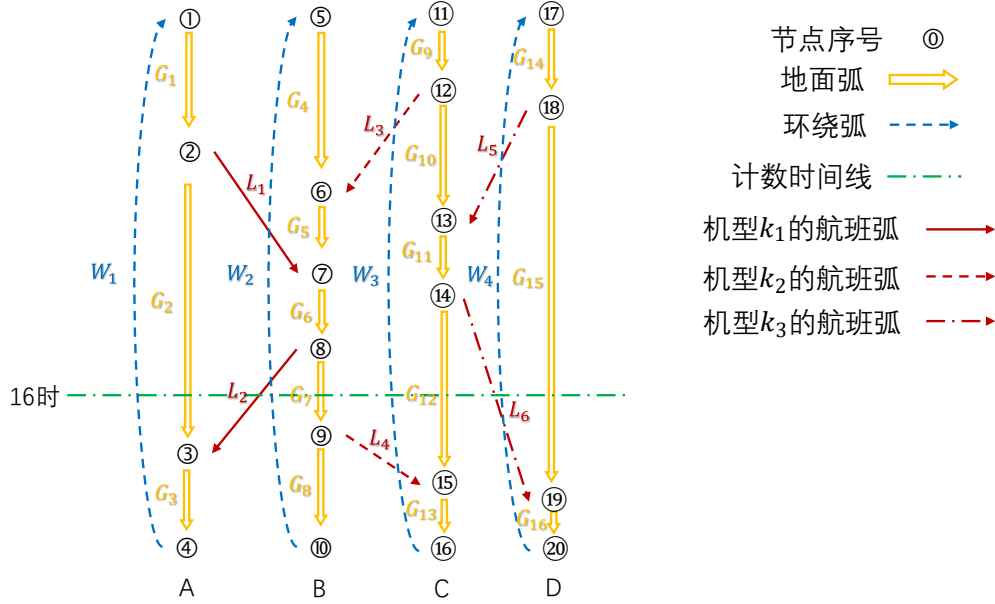


图 11 时空网络的航班拓扑图

在图中，我们分别对航班弧、地面弧与环绕弧命名，例如机型 $k_1$ 的航班②→⑦为航班弧 $L_1$ ，地面弧①→②为 $G_1$ ，机场 A 的环绕弧④→①为 $W_1$ 。

根据图 11 我们在如下表格中给出时空网络模型的参数定义，包括数学符号、具体解释和示例。

集合	
$K = \{k\}$	机队 $K$ 由机型 $k$ 组成，例如示例中的 $K = \{k_1, k_2, k_3\}$
$L = \{l\}$	$L$ 表示航班集合，由航班 $l$ 组成，例如示例中的 $L = \{L_1, L_2, L_3, L_4, L_5, L_6\}$
$L_m$	$L_m$ 表示已指派机型的航班集合即必要航班集合，在本例中我们假定必须执行的航班为 $L_m = \{L_1\}$ 。
$L_o$	$L_o$ 表示可选航班集合，在本例中我们假定不一定执行的航班为 $L_o = \{L_2, L_3, L_4, L_5, L_6\}$ 。
$N_k = \{n\}$	$N_k$ 表示机型 $k$ 的时空网络中的节点集合，由节点 $n$ 组成，例如在本例中机型 $k_1$ 的节点集合为 $N_1 = \{1, 2, 7, 8, 3, 4\}$
$L_{n^+}$	$L_{n^+}$ 表示到达节点 $n$ 的航班集合，例如图中到达节点⑦的航班集合为 $L_{7^+} = \{L_1\}$
$L_{n^-}$	$L_{n^-}$ 表示从节点 $n$ 出发的航班集合，例如图中从节点⑧出发的航班集合为 $L_{8^+} = \{L_2\}$
$L^C$	$L^C$ 表示穿越计数时间线的航班集合，计数时间线可以为一天中的任意时刻，本例中 $L^C = \{L_2, L_6\}$
$N_k^C$	$N_k^C$ 表示时空网络中机型 $k$ 从地面弧上进入节点 $n$ 时并穿越计数时间线的节点集合，例如机型 $k_2$ 的 $N_1^C = \{9\}$

常量	
$M_k$	机队 $K$ 的每个机型 $k$ 的可用飞机数量为 $M_k$ , 本例中我们假设 $M_{k_1} = M_{k_2} = M_{k_3} = 1$
变量	
$c_k$	机队 $K$ 中使用每架飞机的成本
$p_{lk}$	运营由机型 $k$ 执行的航班 $l$ 的收益
$q_{n^+}$	$q_{n^+}$ 表示由地面弧上进入节点 $n$ 的飞机数量即仍在地面上还未执行航班任务的飞机, 其中 $n \in N_k, k \in K$ ; 例如对于机型 $k_2$ , $q_{7^+} = q_{8^+} = q_{9^+} = 1$ (即该机型 $k_2$ )
$q_{n^-}$	$q_{n^-}$ 表示由地面弧上离开节点 $n$ 的飞机数量, 即地面上准备执行航班任务的飞机, 其中 $n \in N_k, k \in K$ ;
$q_{w_k^+}$	$q_{w_k^+}$ 表示由环绕弧进入机场起始节点 $n$ 的飞机数量, 即前一天在机场过夜, 准备执行第二天的航班任务的飞机, 其中 $k \in K$ ; 例如对于机型 $k_1$ 来说, $q_{w_{k_1}^+} = 1$
$q_{w_k^-}$	$q_{w_k^-}$ 表示离开机场末端节点 $n$ 进入环绕弧的飞机数量, 即完成当天执飞任务, 并在该机场过夜的飞机, 其中 $k \in K$ ; 例如对于机型 $k_1$ 来说, $q_{w_{k_1}^-} = 1$
决策变量	
$x_{lk} \in \{0,1\}$	当机队 $k$ 分配到航班 $l$ 执飞时, $x_{lk} = 1$ ; 否则, $x_{lk} = 0$

表 时空网络模型的参数表

针对上述四种参数, 我们在下一小节提出目标函数与约束条件。值得注意的是,  $L_m$  和  $L_o$  是我们结合航线网络设计所提出来的必要航班集合和可选航班集合, 为了优化模型使得其符合实际航班规划的设计。

而  $L_{n^+}$  和  $L_{n^-}$  两种集合的提出, 是为了计算计数时刻空中飞机数量。同样的,  $q_{n^+}$  和  $q_{n^-}$  是为了计算出计数时刻地面上的飞机数量, 以此来满足流量平衡约束。而  $q_{w_k^+}$  和  $q_{w_k^-}$  是为了满足环绕弧的流量守恒, 使得调度飞机平衡, 各种机型能返回原机场来保证航班计划的每日重复执行。

## 5.2. 时空网络优化模型的目标函数与约束条件

接下来, 我们在图 11 的基础上, 给出以下目标函数与约束条件。

目标函数的研究, 仍然是保证收益最大化和成本最小化, 所以依然可以用下式表示。

目标函数 =  $Max$  (当天所有航段的收益 - 当天调度所使用的飞机成本)

这里所有的航段收益我们可以用决策变量来表示, 每个航段决策变量与收益变量的乘积得到。而调度所使用的飞机成本, 这里我们需要考虑到时空网络法统计飞机时, 不仅需要包含正在执行航班的空中飞机, 还要包含机场地面弧上准备执飞的地面航空器。因此, 我们需要用计数时间线帮助我们统计飞机的数量。在图 11 中, 我们采用 16 时作为计数时间线。所以, 空中飞机数量就用穿过计数时间线的航班数量表示; 而地面飞机的数量, 我们用从地面弧上进入节点并穿越计数时间线的节点集合表示。在本例中, 空中飞机与地面飞机的具体计算如下表所示。

空中飞机数量			地面飞机数量		
穿越计数时间线的航班弧 $L$	航班机型 $k$	决策变量 $x_{lk}$	穿越计数时间线的地面弧 $G$	进入节点 $n$	变量 $q_{n^+}$
$L_2$	$k_1$	$x_{L_2 k_1}$	$G_2$	③	$q_{3^+} = 0$
$L_6$	$k_3$	$x_{L_6 k_3}$	$G_7$	⑨	$q_{9^+} = 1$



	$G_{12}$	⑮	$q_{12^+} = 0$
	$G_{15}$	⑲	$q_{15^+} = 0$

将上述表格化简，并采用我们参数表中定义的符号表示，可以得到空中飞机数量可以用以下公式(13)表示。

$$\sum_{l \in L^C} x_{lk} \quad (13)$$

其中航班为穿越计数时间线的航班集合，决策变量为二元变量，若机型 $k$ 覆盖 $l$ 航段，决策变量为1。

根据表格，地面飞机数量可以用进入节点时穿越计数时间线的地面弧表示，其中变量 $q_{n^+} \geq 0$ ，如下公式(14)

$$\sum_{n \in N_k^C} q_{n^+} \quad (14)$$

而空中飞机加上地面飞机的数量就为当天调度所使用的飞机数量，乘上成本变量为飞机调度成本，收益则用收益变量乘上决策变量。

综上所述，可得目标函数公式为(15)，其中左侧为收益，右侧为成本，使目标函数最大化。

$$\text{Max} \quad \sum_{l \in L} \sum_{k \in K} p_{lk} x_{lk} - \left( \sum_{l \in L^C} \sum_{k \in K} c_k x_{lk} + \sum_{n \in N_k^C} \sum_{k \in K} c_k q_{n^+} \right) \quad (15)$$

针对机型指派问题，我们通常所需要满足的就是五种类型的联合约束：航班覆盖约束、设备连续性约束、流量平衡约束、飞机调度平衡约束和可用飞机最大数量约束。

在这里，设备连续性约束的目标是为了保证网络流量平衡，而连接网络法中只需要对每个节点进行流量平衡约束即可以保证该时空网络图的流量平衡，因此我们仅需要满足四种类型的约束。

而为了对该时空网络图进行航线网络设计，让航空公司在原有航班计划上进行修改，我们将航班覆盖约束扩展为可选航班约束与必要航班约束。

必要航班约束要求将航班集合分为两部分，一部分是必要集合 $L_m$ ，另一部分是可选集合 $L_o$ 。在本例中，假定 $L_m = \{L_1\}$ ，我们可将必要航班约束表示如下表格 8，左侧连接为加号。

**表格 8 必要航班约束**

对应航班	左侧（需指派机型的航班）		符号	右侧（航班需分配机型）
	机型 $k_1$	机型 $k_2$		

$L_1$	$x_{L_1 k_1}$	$x_{L_1 k_2}$	=	1
-------	---------------	---------------	---	---

将上述表格化简后可得必要航班约束(16)

$$\sum_{k \in K} x_{lk} = 1 \quad \forall l \in L_m \quad (16)$$

同样的, 本例假定可选航班集合  $L_o = \{L_2, L_3, L_4, L_5, L_6\}$ , 我们可以将其表示为下表所示。

**表格 9 可选航班约束**

对应航班	左侧（不一定指派机型的航班）		符号	右侧（航班是否分配机型）
	机型 $k_1$	机型 $k_2$		
$L_2$	$x_{L_2 k_1}$	$x_{L_2 k_2}$	$\leq$	1
$L_3$	$x_{L_3 k_1}$	$x_{L_3 k_2}$	$\leq$	1
$L_4$	$x_{L_4 k_1}$	$x_{L_4 k_2}$	$\leq$	1
$L_5$	$x_{L_5 k_1}$	$x_{L_5 k_2}$	$\leq$	1
$L_6$	$x_{L_6 k_1}$	$x_{L_6 k_2}$	$\leq$	1

将上述表格化简后可得可选航班约束(17), 表示该航段不一定执飞, 帮助航空公司根据自身资源来调整机型分配策略。

$$\sum_{k \in K} x_{lk} \leq 1 \quad \forall l \in L_o \quad (17)$$

我们对节点流量平衡的分析, 发现每个活动节点都有输入流量和输出流量, 大致可以表示为  $L_1 + G_1 + W_1 = L_2 + G_2 + W_2$ , 而为了使网络流量平衡, 我们需要保证每个节点的流量平衡。因为不是所有节点都有环绕弧, 因此我们需要先保证每个节点上飞机流量的平衡, 再约束环绕弧上的流量平衡。

而约束节点流量平衡的方法, 就是使得每个节点进港和离港的飞机数量相等。同样的, 我们需要保证进入节点时空中飞机数量加上地面飞机数量等于离开节点时空中飞机数量加上地面飞机数量。由于需要保证每个节点的流量守恒, 我们不需要用计数时间线来统计已经调度和分配机型的航班, 只需要统计对于该节点的飞机流量守恒即可。因此, 在本例中, 由于节点数量太多, 对该例我们仅表示节点  $\{2, 3, 6, 9, 13, 19\}$  作为举例, 流量平衡约束可以表示为如下表所示。

表格 10 节点流量平衡约束

对应 航班	对应 节点	左侧（进入节点）		符号	右侧（离开节点）	
		空中流量	地面流量		空中流量	地面流量
$L_1$	②		$q_{2^+} = 1$	=	$x_{L_1 k_1}$	$q_{2^-} = 0$
$L_2$	③	$x_{L_2 k_1}$	$q_{3^+} = 0$	=		$q_{3^-} = 0$
$L_3$	⑥	$x_{L_3 k_2}$	$q_{4^+} = 0$	=		$q_{4^-} = 1$
$L_4$	⑨		$q_{9^+} = 1$	=	$x_{L_4 k_2}$	$q_{9^-} = 0$
$L_5$	⑬	$x_{L_5 k_3}$	$q_{13^+} = 0$	=		$q_{13^-} = 1$
$L_6$	⑰	$x_{L_6 k_3}$	$q_{19^+} = 0$	=		$q_{19^-} = 1$

将上述表格化简后可得约束条件(18)节点流量平衡，保证每个节点的流量平衡。

$$\sum_{l \in L_{n^+}} x_{lk} + q_{n^+} = \sum_{l \in L_{n^-}} x_{lk} + q_{n^-} \quad \forall n \in N_k, \forall k \in K \quad (18)$$

该约束条件保证了每个节点的流量守恒，但第 $N$ 天的时空网络图中，我们还需要让同一机场的起始环绕弧和结束环绕弧的流量值必须相同。飞机调度平衡约束，保证了航班计划可以每天重复循环执行，减少航空公司的网络设计难度。因此，我们需要让每个环绕弧的起始节点流量与环绕弧末端节点的流量相同，即让在该机场准备过夜的飞机数量等于第二天离开机场执飞的飞机数量，如下表所示。

表格 11 飞机调度平衡

对应 机场	机 型	对应 节点	左侧（过夜后准备执 飞的飞机）	符 号	对应 节点	右侧（准备过夜 的飞机）
$A$	$k_1$	①	$q_{w_{k_1}^+} = 1$	=	④	$q_{w_{k_1}^-} = 1$
$B$		⑤	$q_{w_k^+} = 0$	=	⑩	$q_{w_k^+} = 0$
$C$	$k_2$	⑪	$q_{w_{k_2}^+} = 1$	=	⑰	$q_{w_{k_2}^-} = 1$
$D$	$k_3$	⑰	$q_{w_{k_3}^+} = 1$	=	⑳	$q_{w_{k_3}^-} = 1$

将上述表格化简后可得约束条件(19)飞机调度平衡，保证航班计划可重复执行。

$$q_{w_k^+} = q_{w_k^-} \quad \forall k \in K \quad (19)$$

除此之外, 我们还需要考虑飞机数量约束, 使得所调用的飞机数量对于该机型来说不超过最大机型数量。同样的, 我们在考虑目标函数成本的时候, 计算了空中飞机数量和地面飞机数量, 用计数时间线算出了该时刻的空中与地面飞机数量, 因此我们根据公式(13)和(14)列出飞机数量约束(20)。

$$\sum_{l \in L^C} x_{lk} + \sum_{n \in N_k^C} q_{n^+} \leq M_k \quad \forall k \in K \quad (20)$$

此外, 还有决策变量约束(21), 保证其为 0-1 变量。

$$x_{lk} \in \{0,1\} \quad \forall l \in L, \forall k \in K \quad (21)$$

变量约束(22), 让地面上进入节点的飞机数量和离开节点的飞机数量不小于 0。

$$y_{n^+}, y_{n^-} \geq 0 \quad \forall n \in N_k, \forall k \in K \quad (22)$$

综上所述, 我们将时空网络优化模型的目标函数与约束条件归纳如下。

$$\text{Max} \quad \sum_{l \in L} \sum_{k \in K} p_{lk} x_{lk} - \left( \sum_{l \in L^C} \sum_{k \in K} c_k x_{lk} + \sum_{n \in N_k^C} \sum_{k \in K} c_k q_{n^+} \right) \quad (23)$$

s. t.

$$\sum_{k \in K} x_{lk} = 1 \quad \forall l \in L_m \quad (24)$$

$$\sum_{k \in K} x_{lk} \leq 1 \quad \forall l \in L_o \quad (25)$$

$$\sum_{l \in L_{n^+}} x_{lk} + q_{n^+} = \sum_{l \in L_{n^-}} x_{lk} + q_{n^-} \quad \forall n \in N_k, \forall k \in K \quad (26)$$

$$q_{w_k^+} = q_{w_k^-} \quad \forall k \in K \quad (27)$$

$$\sum_{l \in L^C} x_{lk} + \sum_{n \in N_k^C} q_{n^+} \leq M_k \quad \forall k \in K \quad (28)$$

$$x_{lk} \in \{0,1\} \quad \forall l \in L, \forall k \in K \quad (29)$$

$$y_{n^+}, y_{n^-} \geq 0 \quad \forall n \in N_k, \forall k \in K \quad (30)$$

## 6. 实例数据

数据均来自全日空航空公司 (All Nippon Airways Co., Ltd.) 于 2023 年所制定的中国航线夏季时刻表, 机场代码均采用 IATA 所规定的三字代码。由于本文仅简单考虑经济因素, 因此, 机型的运营成本和运营收益大致用 CASM (可用座公里成本) 和 RASM (可用座公里收入) 大致代换粗略得到。

该算例共涉及 24 个航班，70 个节点，4 种机型，8 个机场，全日空航空公司夏季某日的航班计划表如下所示。

**表格 12 全日空航空公司航班计划表**

编号	航班号	起始机场	起飞时间	目的机场	到达时间	是否为必要航班
1	NH969	HND	10:05	SHA	12:15	是
2	NH970	SHA	13:35	HND	17:25	是
3	NH967	HND	21:30	PVG	23:35	是
4	NH968	PVG	1:35	HND	5:30	是
5	NH919	NRT	9:20	PVG	11:40	是
6	NH920	PVG	13:05	NRT	16:55	是
7	NH973	KIX	9:00	PVG	10:35	否
8	NH974	PVG	11:35	KIX	14:55	否
9	NH961	HND	8:55	PEK	12:00	是
10	NH962	PEK	15:10	HND	19:45	是
11	NH955	NRT	9:00	PEK	12:05	否
12	NH956	PEK	14:15	NRT	18:40	否
13	NH931	NRT	11:20	SZX	15:10	否
14	NH932	SZX	17:00	NRT	22:15	否
15	NH965	HND	11:30	SZX	15:10	是
16	NH966	SZX	17:00	HND	22:20	是
17	NH927	NRT	9:45	TAO	12:30	否
18	NH928	TAO	13:30	NRT	17:35	否
19	NH933	NRT	9:10	CAN	13:05	是
20	NH934	CAN	14:15	NRT	19:45	是
21	NH929	NRT	10:05	HGH	12:50	否
22	NH930	HGH	13:40	NRT	17:55	否
23	NH903	NRT	10:05	DLC	12:15	否
24	NH904	DLC	13:15	NRT	17:15	否

如上表所示，给出了某日航班计划的具体时间表，航班出发和到达时刻均有记录。其中我们将航班分为必要航班和可选航班，必要航班的航线为航空公司必须执行的干线，即在考虑经济效益的情况下，需每日执行的航线。而可选航班为航空公司所设计的支线，不需要每日执行，在综合考虑机型因素、经济因素后对

候选航班进行选择。最终，我们对这些航线进行合理的机型分配。根据上表，我们可用画出时空网络的航班计划图，如下图所示。

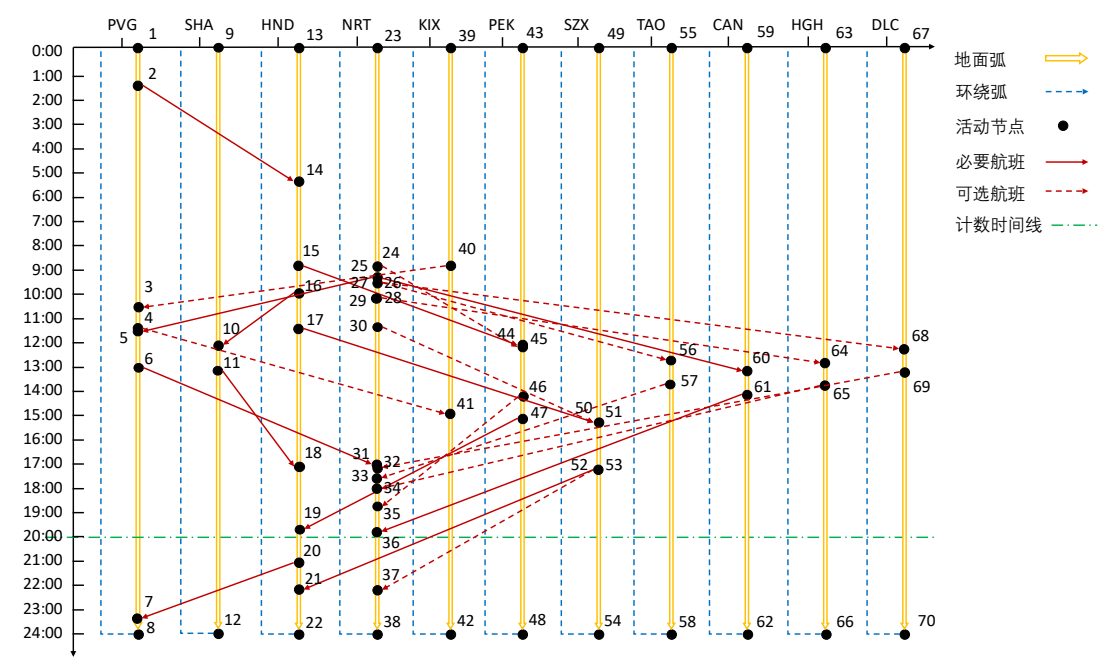


图 12 时空网络航班计划图

如上图所示，该图显示了采用时空网络法表示的航班计划图，活动节点表示航班的进港和出港活动，有向箭头表示航班的起飞和到达方向，网络图中的节点均用序号表示出来包括环绕弧的起始节点与末端节点，表示机型到达该机场完成航班任务后在机场过夜。计数时间线用来计算空中与地面的飞机数量，其可以设计为任意时刻，在本例中我们将其设定为 20:00。因此，我们可以看见穿越计数时间线的航班编号为 14 和 16，在此计为空中飞机的数量，同理通过计算穿越计数时间线的地面弧可得地面飞机的数量。

通过收集数据可得，全日空航空公司在执行 2023 年夏季航线时共采用以下 4 种机型：波音 787-8、波音 787-9、波音 763 和空客 A320（以下均用缩写代替），我们假定其可执飞机队的机型数量，并根据座位数推测其使用机型成本与收益。最终得到算例所用到的相关实验数据如下表所示。

表格 13 时空网络实验数据表

参数名	实验数据	
机队 <i>K</i> 的组成和数量	机型	数量
	$k_1$ : B787-800	4
	$k_2$ : B787-900	6
	$k_3$ : B767-300	5

	$k_4$ : A320	2			
航班集合 $L$	以航班编号命名为 $L_1 \sim L_{24}$				
可选航班集合 $L_o$	$\{L_7, L_8, L_{11}, L_{12}, L_{13}, L_{14}, L_{17}, L_{18}, L_{21}, L_{22}, L_{23}, L_{24}\}$				
必要航班集合 $L_m$	除可选航班集合外的其他航班集合				
节点集合 $N_k$	本例中共有 70 个节点标号				
计数时间线	设定计数时刻为 20:00				
使用一架飞机的成本 $c_k$	使用机型 $k_1$ : 12000				
	使用机型 $k_2$ : 13000				
	使用机型 $k_3$ : 17000				
	使用机型 $k_4$ : 21000				
不同航线采用不同机型的收益 $p_{lk}$	航线	不同机型的 $p_{lk}$			
		$k_1$	$k_2$	$k_3$	$k_4$
	$\{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8\}$	21612	29138	34676	26646
	$\{L_9, L_{10}, L_{11}, L_{12}\}$	25032	33750	40165	30863
	$\{L_{13}, L_{14}, L_{15}, L_{16}\}$	32292	43538	51813	39814
	$\{L_{17}, L_{18}\}$	29493	39764	47323	36364
	$\{L_{19}, L_{20}\}$	35067	47279	56265	43235
	$\{L_{21}, L_{22}\}$	22724	30638	36461	28017
	$\{L_{23}, L_{24}\}$	25116	33863	40299	30967

在上表中的航线收益是根据航线的航程公里数和机型的 RASM(可用座公里收入) 以及机型的座位数来综合计算得出的, 由于本文仅考虑经济因素, 假设所有机型满座且无旅客溢出成本等, 计算公式如下。

$$\text{航线收益} = \text{可用座公里收入} \times \text{座位数} \times \text{航程公里数}$$

要求调用商用求解器求解, 将上述模型参数带入求解。

## 7. 课设任务

- (1) 分析问题建立模型
- (2) 编写 Gurobi/Lindo/Lingo 求解程序
- (3) 运算求解
- (4) 结果评估分析
- (5) 撰写课程设计报告

## 8. 课程设计报告格式要求

提交课程设计报告，报告须包括问题的描述、数学模型、求解和结果评估、决策建议和参考文献等五部分组成。

课程设计报告格式要求见附件 4 设计报告模版。提交的正式报告应当去掉“附件 3 设计报告模板”字样。



---

## 附件

### 附件 1 Gurobi 安装与使用方法

许可证的申请方法为：<http://www.gurobi.cn/NewsView1.Asp?id=4>。

Anaconda 中安装使用 Gurobi 的介绍：

[https://www.gurobi.com/documentation/9.5/quickstart\\_linux/cs\\_anaconda\\_and\\_grb\\_conda\\_.html](https://www.gurobi.com/documentation/9.5/quickstart_linux/cs_anaconda_and_grb_conda_.html)。

Gurobi 的用户手册：<https://www.gurobi.com/documentation/10.0/refman/refman.html>

## 附件 2 Lindo/Lingo 功能与使用方法

### 一、前言

Lindo/Lingo 是优化建模和求解的商用软件，是一个公司的姊妹产品。LINDO 软件主要用于求解线性规划、整数规划、二次规划问题。LINGO 用于求解线性规划、整数规划以及非线性规划问题。Lindo 建模简单，同时也就只能解决较为简单的问题，Lingo 的建模功能比较强大，可以解决 Lindo 能解决的所有问题，也能解决 Lindo 不能解决的问题，是解决工程实际问题的有用工具。

Lindo 和 Lingo 建模程序结构的主要差异如下：

#### 1、Lindo:

MAX 目标函数表达

ST

变量约束 1

变量约束 2

变量约束 3

END

#### 2、Lingo:

MAX=目标函数表达;

变量约束 1;

变量约束 2;

变量约束 3;

从上述各表达式可以看出，在 Lingo 中每个语句后面必须以分号结束，包括以 ! 开头的注释语句。

在课程设计中，主要利用 Lingo 软件求解模型，为此，下面详细介绍 Lingo 软件的应用。

Lingo 是一个利用线性规划和非线性规划来简洁地阐述、解决和分析复杂问题的简便工具。其特点是程序执行速度很快，易于输入、修改、求解和分析一个数学规划问题，因此 Lingo 在教育、科研和工业界得到了广泛应用。

Lingo 内置了一种建立最优化模型的语言，可以简便地表达大规模问题，利用 Lingo 高效的求解器可快速求解并分析结果。

### 二、LINGO 快速入门

在 windows 下开始运行 LINGO 系统时，会得到如图 1 所示的一个窗口：

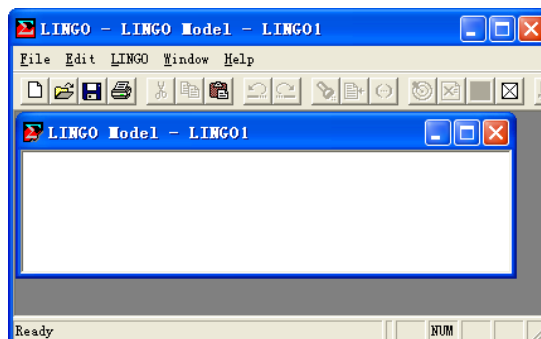


图 1 Lingo 的界面

外层是主框架窗口，包含了所有菜单命令和工具条，其它所有的窗口将被包含在主窗口之内。在主窗口内标题为 LINGO Model-LINGO1 的窗口是 LINGO 的默认模型窗口，建立的

模型都要在该窗口内编码实现。下面举两个例子。

例 2.1 如何在 LINGO 中求解如下的 LP 问题：

$$\begin{aligned} \min \quad & 2x_1 + 3x_2 \\ \text{s.t.} \quad & \\ & x_1 + x_2 \geq 350 \\ & x_1 \geq 100 \\ & 2x_1 + x_2 \leq 600 \\ & x_1, x_2 \geq 0 \end{aligned}$$

在模型窗口中输入如下代码：

```
min=2*x1+3*x2;  
x1+x2>=350;  
x1>=100;  
2*x1+x2<=600;
```

然后点击工具条上的按钮  即可得到该问题的最优解。

### 三、LINGO 中的集

对实际问题建模的时候，总会遇到一群或多群相联系的对象，比如工厂、消费者群体、交通工具和雇工等等。LINGO 允许把这些相联系的对象聚合成集（sets）。一旦把对象聚合成集，就可以利用集来最大限度的发挥 LINGO 建模语言的优势。

#### 3.1 为什么使用集

集是 LINGO 建模语言的基础，是程序设计最强有力的基本构件。借助于集，能够用一个单一的、长的、简明的复合公式表示一系列相似的约束，从而可以快速方便地表达规模较大的模型。

#### 3.2 什么是集

集是一群相联系的对象，这些对象也称为集的成员。一个集可能是一系列产品、卡车或雇员。每个集成员可能有一个或多个与之有关联的特征，我们把这些特征称为属性。属性值可以预先给定，也可以是未知的，有待于 LINGO 求解。例如，产品集中的每个产品可以有一个价格属性；卡车集中的每辆卡车可以有一个牵引力属性；雇员集中的每位雇员可以有一个薪水属性，也可以有一个生日属性等等。

LINGO 有两种类型的集：原始集(primitive set)和派生集(derived set)。

一个原始集是由一些最基本的对象组成的。

一个派生集是用一个或多个其它集来定义的，也就是说，它的成员来自于其它已存在的集。

#### 3.3 模型的集部分

集部分是 LINGO 模型的一个可选部分。在 LINGO 模型中使用集之前，必须在集部分事先定义。集部分以关键字“sets:”开始，以“endsets”结束。一个模型可以没有集部分，或有一个简单的集部分，或多个集部分。一个集部分可以放置于模型的任何地方，但是一个集及其属性在模型约束被引用之前必须定义了它们。

##### 3.3.1 定义原始集

为了定义一个原始集，必须详细声明：

·集的名字

·可选，集的成员

·可选，集成员的属性

定义一个原始集，用下面的语法：

```
setname[/member_list/][:attribute_list];
```

注意：用“[]”表示该部分内容可选。下同，不再赘述。

Setname 是你选择的用来标记集的名字，最好具有较强的可读性。集名字必须严格符合标准命名规则：以拉丁字母或下划线（\_）为首字符，其后由拉丁字母（A—Z）、下划线、阿拉伯数字（0，1，...，9）组成的总长度不超过 32 个字符的字符串，且不区分大小写。

注意：该命名规则同样适用于集成员名和属性名等的命名。

Member\_list 是集成员列表。如果集成员放在集定义中，那么对它们可采取显式罗列和隐式罗列两种方式。如果集成员不放在集定义中，那么可以在随后的数据部分定义它们。

① 当显式罗列成员时，必须为每个成员输入一个不同的名字，中间用空格或逗号隔开，允许混合使用。

例 3.1 可以定义一个名为 students 的原始集，它具有成员 John、Jill、Rose 和 Mike，属性有 sex 和 age:

sets:

```
students/John,Jill, Rose,Mike/: sex, age;
```

endsets

② 当隐式罗列成员时，不必罗列出每个集成员。可采用如下语法：

```
setname/member1..memberN/[: attribute_list];
```

这里的 member1 是集的第一个成员名，memberN 是集的最末一个成员名。LINGO 将自动产生中间的所有成员名。LINGO 也接受一些特定的首成员名和末成员名，用于创建一些特殊的集。列表如表 1 所示。

表 1 常用隐式成员列表格式

隐式成员列表格式	示例	所产生集成员
1..n	1..5	1,2,3,4,5
StringM..StringN	Car2..car14	Car2,Car3,Car4,...,Car14
DayM..DayN	Mon..Fri	Mon,Tue,Wed,Thu,Fri
MonthM..MonthN	Oct..Jan	Oct,Nov,Dec,Jan
MonthYearM..MonthYearN	Oct2001..Jan2002	Oct2001,Nov2001,Dec2001,Jan2002

③ 集成员不放在集定义中，而在随后的数据部分来定义。

例 3.2

!集部分;

sets:

```
students:sex,age;
```

endsets

!数据部分;

data:

```
students,sex,age= John 1 16
```

---

```
Jill 0 14
Rose 0 17
Mike 1 13;
```

**enddata**

*注意：开头用感叹号 (!)，末尾用分号 (;) 表示注释，可跨多行。*

在集部分只定义了一个集 **students**，并未指定成员。在数据部分罗列了集成员 **John**、**Jill**、**Rose** 和 **Mike**，并对属性 **sex** 和 **age** 分别给出了值。

集成员无论用何种字符标记，它的索引都是从 1 开始连续计数。在 **attribute\_list** 可以指定一个或多个集成员的属性，属性之间必须用逗号隔开。

### 3.3.2 定义派生集

为了定义一个派生集，必须详细声明：

- 集的名字
- 父集的名字
- 可选，集成员
- 可选，集成员的属性

可用下面的语法定义一个派生集：

```
setname(parent_set_list)/[member_list]/[:attribute_list];
```

**setname** 是集的名字。**parent\_set\_list** 是已定义的集的列表，多个时必须用逗号隔开。如果没有指定成员列表，那么 LINGO 会自动创建父集成员的所有组合作为派生集的成员。派生集的父集既可以是原始集，也可以是其它的派生集。

例 3.3

**sets:**

```
product/A B/;
machine/M N/;
week/1..2/;
allowed(product,machine,week):x;
```

**endsets**

LINGO 生成了三个父集的所有组合共八组作为 **allowed** 集的成员。列表如下：

编号	成员
1	(A,M,1)
2	(A,M,2)
2	(A,N,1)
3	(A,N,2)
4	(B,M,1)
5	(B,M,2)
6	(B,N,1)
7	(B,N,2)

成员列表被忽略时，派生集成员由父集成员所有的组合构成，这样的派生集成为稠密集。如果限制派生集的成员，使它成为父集成员所有组合构成的集合的一个子集，这样的派生集成为稀疏集。同原始集一样，派生集成员的声明也可以放在数据部分。一个派生集的成员列表有两种方式生成：①显式罗列；②设置成员资格过滤器。当采用方式①时，必须显式罗列所有要包含在派生集中的成员，并且罗列的每个成员必须属于稠密集。使用前面的例子，

---

显式罗列派生集的成员：

```
allowed(product,machine,week)/A M 1,A N 2,B N 1/;
```

如果需要生成一个大的、稀疏的集，那么显式罗列就很讨厌。幸运的是许多稀疏集的成员都满足一些条件以和非成员相区分。我们可以把这些逻辑条件看作过滤器，在 LINGO 生成派生集的成员时能把逻辑条件为假的成员从稠密集中过滤掉。

#### 例 3.4

sets:

```
!学生集：性别属性 sex，1 表示男性，0 表示女性；年龄属性 age. ；  
students/John,Jill,Rose,Mike/:sex,age;  
!男学生和女学生的联系集：友好程度属性 friend，[0，1]之间的数。 ；  
linkmf(students,students)|sex(&1) #eq# 1 #and# sex(&2) #eq# 0: friend;  
!男学生和女学生的友好程度大于 0.5 的集；  
linkmf2(linkmf) | friend(&1,&2) #ge# 0.5 : x;
```

endsets

data:

```
sex,age = 1 16  
          0 14  
          0 17  
          0 13;  
friend = 0.3 0.5 0.6;
```

enddata

用竖线 (|) 来标记一个成员资格过滤器的开始。#eq#是逻辑运算符，用来判断是否“相等”，可参考§4. &1 可看作派生集的第 1 个原始父集的索引，它取遍该原始父集的所有成员；&2 可看作派生集的第 2 个原始父集的索引，它取遍该原始父集的所有成员；&3, &4, ……，以此类推。注意如果派生集 B 的父集是其它集的派生集 A，那么上面所说的原始父集是集 A 向前回溯到最终的原始集，其顺序保持不变，并且派生集 A 的过滤器对派生集 B 仍然有效。因此，派生集的索引个数是最终原始父集的个数，索引的取值是从原始父集到当前派生集所作限制的总和。

总的来说，LINGO 可识别的集只有两种类型：原始集和派生集。不同集类型的关系见图 2。

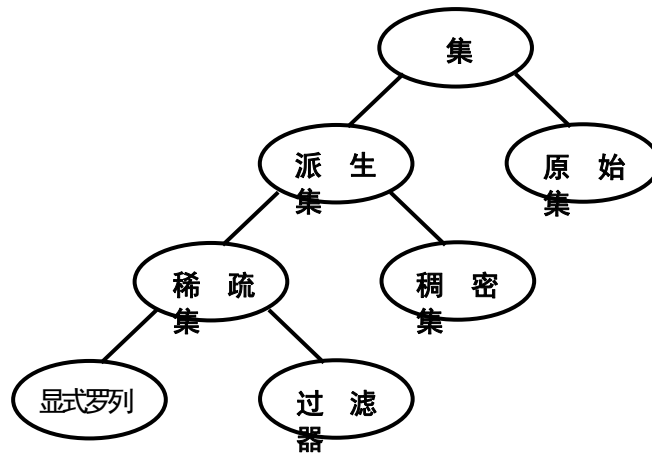


图 2 LINGO 集 类  
型

#### 四、模型的数据部分和初始部分

在处理模型的数据时，需要为集指派一些成员并且在 LINGO 求解模型之前为集的某些属性指定值。为此，LINGO 为用户提供了两个可选部分：输入集成员和数据的数据部分（Data Section）和为决策变量设置初始值的初始部分（Init Section）。

##### 4.1 模型的数据部分

###### 4.1.1 数据部分入门

数据部分提供了模型相对静止部分和数据分离的可能性。显然，这对模型的维护和维数的缩放非常便利。

数据部分以关键字“data:”开始，以关键字“enddata”结束。在这里，可以指定集成员、集的属性。其语法如下：

`object_list = value_list;`

对象列（object\_list）包含要指定值的属性名、要设置集成员的集名，用逗号或空格隔开。一个对象列中至多有一个集名，而属性名可以有任意多。如果对象列中有多个属性名，那么它们的类型必须一致。如果对象列中有一个集名，那么对象列中所有的属性的类型就是这个集。

数值列（value\_list）包含要分配给对象列中的对象的值，用逗号或空格隔开。注意属性值的个数必须等于集成员的个数。

###### 4.1.2 参数

在数据部分也可以指定一些标量变量（scalar variables）。当一个标量变量在数据部分确定时，称之为参数。看一例，假设模型中用利率 8.5%作为一个参数，就可以象下面一样输入一个利率作为参数。

例 4.1

data:

interest\_rate = 0.085;

enddata

也可以同时指定多个参数。

例 4.2

data:

interest\_rate,inflation\_rate = 0.085,0.03;

enddata

### 4.1.3 实时数据处理

在某些情况,对于模型中的某些数据并不是定值。譬如模型中有一个通货膨胀率的参数,我们想在 2%至 6%范围内,对不同的值求解模型,来观察模型的结果对通货膨胀的依赖的敏感性。把这种情况称为实时数据处理(what if analysis)。LINGO 有一个特征可方便地做到这件事:在本该放数的地方输入一个问号(?)。

例 4.3

data:

```
interest_rate,inflation_rate = .085  ?;
```

enddata

每一次求解模型时,LINGO 都会提示为参数 inflation\_rate 输入一个值。在 WINDOWS 操作系统下,将会接收到一个类似图 3 的对话框。

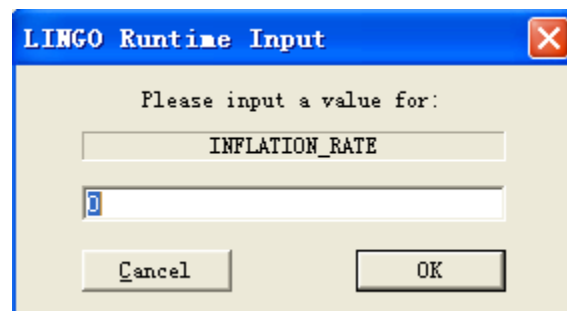


图 3 参数实时输入

直接输入一个值再点击 OK 按钮,LINGO 就会把输入的值指定给 inflation\_rate,然后继续求解模型。

除了参数之外,也可以实时输入集的属性值,但不允许实时输入集成员名。

### 4.1.4 指定属性为一个值

可以在数据声明的右边输入一个值来把所有成员的这个属性指定为一个值。看下面的例子。

例 4.4

sets:

```
days /MO,TU,WE,TH,FR,SA,SU/:needs;
```

endsets

data:

```
needs = 20;
```

enddata

LINGO 将 20 赋给 days 集的所有成员的 needs 属性。对于多个属性的情形,见下例。

例 4.5

sets:

```
days /MO,TU,WE,TH,FR,SA,SU/:needs,cost;
```

endsets

data:

```
needs cost = 20 100;
```

enddata

### 4.1.5 数据部分的未知数值



---

有时只想为一个集的部分成员的某个属性赋值，而让其余成员的该属性保持未知，以便让 LINGO 去求出它们的最优值。在数据声明中输入两个相连的逗号表示该位置对应的集成员的属性值未知。两个逗号间可以有空格。

#### 4.2 模型的初始部分

初始部分是 LINGO 提供的另一个可选部分。在初始部分中，可以输入初始声明（initialization statement），和数据部分中的数据声明相同。对实际问题建模时，初始部分并不起到描述模型的作用，在初始部分输入的值仅被 LINGO 求解器当作初始点来用，并且仅仅对非线性模型有用。和数据部分指定变量的值不同，LINGO 求解器可以自由改变初始部分初始化的变量的值。

一个初始部分以“init:”开始，以“endinit”结束。初始部分的初始声明规则和数据部分的数据声明规则相同。也就是说，我们可以在声明的左边同时初始化多个集属性，可以把集属性初始化为一个值，可以用问号实现实时数据处理，还可以用逗号指定未知数值。

### 五、LINGO 函数

有了前几节的基础知识，再加上本节的内容，你就能够借助于 LINGO 建立并求解复杂的优化模型了。

LINGO 有 9 种类型的函数：

1. 基本运算符：包括算术运算符、逻辑运算符和关系运算符
2. 数学函数：三角函数和常规的数学函数
3. 金融函数：LINGO 提供的两种金融函数
4. 概率函数：LINGO 提供了大量概率相关的函数
5. 变量界定函数：这类函数用来定义变量的取值范围
6. 集操作函数：这类函数为对集的操作提供帮助
7. 集循环函数：遍历集的元素，执行一定的操作的函数
8. 数据输入输出函数：这类函数允许模型和外部数据源相联系，进行数据的输入输出
9. 辅助函数：各种杂类函数

#### 5.1 基本运算符

这些运算符是非常基本的，甚至可以认为它们是一类函数。事实上，在 LINGO 中它们是非常重要的。

##### 5.1.1 算术运算符

算术运算符是针对数值进行操作的。LINGO 提供了 5 种二元运算符：

^ 乘方  
\* 乘  
/ 除  
+ 加  
- 减

LINGO 唯一的一元算术运算符是取反函数“-”。

这些运算符的优先级由高到底为：

高    - （取反）  
      ^  
      \* /  
低    + -

运算符的运算次序为从左到右按优先级高低来执行。运算的次序可以用圆括号“（）”来改变。

#### 例 5.1 算术运算符示例

$2 - 5 / 3$ ,  $(2 + 4) / 5$  等等。

#### 5.1.2 逻辑运算符

在 LINGO 中，逻辑运算符主要用于集循环函数的条件表达式中，来控制在函数中哪些集成员被包含，哪些被排斥。在创建稀疏集时用在成员资格过滤器中。

LINGO 具有 9 种逻辑运算符：

- #not#** 否定该操作数的逻辑值，**#not#** 是一个一元运算符
- #eq#** 若两个运算数相等，则为 **true**；否则为 **false**
- #ne#** 若两个运算符不相等，则为 **true**；否则为 **false**
- #gt#** 若左边的运算符严格大于右边的运算符，则为 **true**；否则为 **false**
- #ge#** 若左边的运算符大于或等于右边的运算符，则为 **true**；否则为 **false**
- #lt#** 若左边的运算符严格小于右边的运算符，则为 **true**；否则为 **false**
- #le#** 若左边的运算符小于或等于右边的运算符，则为 **true**；否则为 **false**
- #and#** 仅当两个参数都为 **true** 时，结果为 **true**；否则为 **false**
- #or#** 仅当两个参数都为 **false** 时，结果为 **false**；否则为 **true**

这些运算符的优先级由高到低为：

高 **#not#**  
**#eq#** **#ne#** **#gt#** **#ge#** **#lt#** **#le#**  
低 **#and#** **#or#**

#### 例 5.2 逻辑运算符示例

$2 \#gt\# 3 \#and\# 4 \#gt\# 2$ ，其结果为假（0）。

#### 5.1.3 关系运算符

在 LINGO 中，关系运算符主要是被用在模型中，来指定一个表达式的左边是否等于、小于等于、或者大于等于右边，形成模型的一个约束条件。关系运算符与逻辑运算符**#eq#**、**#le#**、**#ge#**截然不同，前者是模型中该关系运算符所指定关系的为真描述，而后者仅仅判断一个该关系是否被满足：满足为真，不满足为假。

LINGO 有三种关系运算符：“=”、“<=”和“>=”。LINGO 中还能用“<”表示小于等于关系，“>”表示大于等于关系。LINGO 并不支持严格小于和严格大于关系运算符。然而，如果需要严格小于和严格大于关系，比如让 A 严格小于 B：

$$A < B$$

那么可以把它变成如下的小于等于表达式：

$$A + \epsilon \leq B$$

这里  $\epsilon$  是一个小的正数，它的值依赖于模型中 A 小于 B 多少才算不等。

下面给出以上三类操作符的优先级：

高 **#not#** - （取反）  
^  
\* /  
+ -

---

#eq# #ne# #gt# #ge# #lt# #le#  
#and# #or#  
低 <= = >=

## 5.2 数学函数

LINGO 提供了大量的标准数学函数：

@abs(x)	返回 x 的绝对值
@sin(x)	返回 x 的正弦值，x 采用弧度制
@cos(x)	返回 x 的余弦值
@tan(x)	返回 x 的正切值
@exp(x)	返回常数 e 的 x 次方
@log(x)	返回 x 的自然对数
@lgm(x)	返回 x 的 gamma 函数的自然对数
@sign(x)	如果 x<0 返回-1；否则，返回 1
@floor(x)	返回 x 的整数部分。当 x>=0 时，返回不超过 x 的最大整数；当 x<0 时，返回不低于 x 的最大整数。
@smax(x1,x2,...,xn)	返回 x1, x2, ..., xn 中的最大值
@smin(x1,x2,...,xn)	返回 x1, x2, ..., xn 中的最小值

## 5.3 金融函数

目前 LINGO 提供了两个金融函数。

1. @fpa(I,n)
2. @fpl(I,n)

## 5.4 概率函数

1. @pbn(p,n,x)

二项分布的累积分布函数。当 n 和（或）x 不是整数时，用线性插值法进行计算。

2. @pcx(n,x)

自由度为 n 的  $\chi^2$  分布的累积分布函数。

3. @peb(a,x)

当到达负荷为 a，服务系统有 x 个服务器且允许无穷排队时的 Erlang 繁忙概率。

4. @pel(a,x)

当到达负荷为 a，服务系统有 x 个服务器且不允许排队时的 Erlang 繁忙概率。

5. @pfd(n,d,x)

自由度为 n 和 d 的 F 分布的累积分布函数。

6. @pfs(a,x,c)

当负荷上限为 a，顾客数为 c，平行服务器数量为 x 时，有限源的 Poisson 服务系统的等待或返修顾客数的期望值。a 是顾客数乘以平均服务时间，再除以平均返修时间。当 c 和（或）x 不是整数时，采用线性插值进行计算。

7. @phg(pop,g,n,x)

超几何（Hypergeometric）分布的累积分布函数。pop 表示产品总数，g 是正品数。从所有产品中任意取出  $n(n \leq \text{pop})$  件。pop, g, n 和 x 都可以是非整数，这时采用线性插值进行计算。

8. @ppl(a,x)

Poisson 分布的线性损失函数，即返回  $\max(0, z-x)$  的期望值，其中随机变量 z 服从均值为 a 的 Poisson 分布。

#### 9. @pps(a,x)

均值为 a 的 Poisson 分布的累积分布函数。当 x 不是整数时，采用线性插值进行计算。

#### 10. @psl(x)

单位正态线性损失函数，即返回  $\max(0, z-x)$  的期望值，其中随机变量 z 服从标准正态分布。

#### 11. @psn(x)

标准正态分布的累积分布函数。

#### 12. @ptd(n,x)

自由度为 n 的 t 分布的累积分布函数。

#### 13. @qrand(seed)

产生服从(0,1)区间的拟随机数。@qrand 只允许在模型的数据部分使用，它将用拟随机数填满集属性。通常，声明一个  $m \times n$  的二维表，m 表示运行实验的次数，n 表示每次实验所需的随机数的个数。在行内，随机数是独立分布的；在行间，随机数是非常均匀的。这些随机数是用“分层取样”的方法产生的。

#### 14. @rand(seed)

返回 0 和 1 间的伪随机数，依赖于指定的种子。典型用法是  $U(I+1)=@rand(U(I))$ 。注意如果 seed 不变，那么产生的随机数也不变。

### 5.5 变量界定函数

变量界定函数实现对变量取值范围的附加限制，共 4 种：

@bin(x)            限制 x 为 0 或 1

@bnd(L,x,U)       限制  $L \leq x \leq U$

@free(x)           取消对变量 x 的默认下界为 0 的限制，即 x 可以取任意实数

@gin(x)           限制 x 为整数

在默认情况下，LINGO 规定变量是非负的，也就是说下界为 0，上界为  $+\infty$ 。@free 取消了默认的下界为 0 的限制，使变量也可以取负值。@bnd 用于设定一个变量的上下界，它也可以取消默认下界为 0 的约束。

### 5.6 集操作函数

LINGO 提供了几个函数帮助处理集。

#### 1. @in(set\_name, primitive\_index\_1 [, primitive\_index\_2, ...])

如果元素在指定集中，返回 1；否则返回 0。

#### 2. @index([set\_name,] primitive\_set\_element)

该函数返回在集 set\_name 中原始集成员 primitive\_set\_element 的索引。如果 set\_name 被忽略，那么 LINGO 将返回与 primitive\_set\_element 匹配的原始集成员的索引。如果找不到，则产生一个错误。

#### 3. @wrap(index, limit)

该函数返回  $j = \text{index} - k * \text{limit}$ ，其中 k 是一个整数，取适当值保证 j 落在区间 [1, limit] 内。该函数相当于 index 模 limit 再加 1。该函数在循环、多阶段计划编制中特别有用。

#### 4. @size(set\_name)

该函数返回集 set\_name 的成员个数。在模型中明确给出集大小时最好使用该函数。它的使用使模型更加数据中立，集大小改变时也更易维护。

### 5.7 集循环函数

集循环函数遍历整个集进行操作。其语法为

@function(setname[(set\_index\_list)][conditional\_qualifier]):

expression\_list);

@function 相应于下面罗列的四个集循环函数之一; setname 是要遍历的集; set\_index\_list 是集索引列表; conditional\_qualifier 是用来限制集循环函数的范围, 当集循环函数遍历集的每个成员时, LINGO 都要对 conditional\_qualifier 进行评价, 若结果为真, 则对该成员执行 @function 操作, 否则跳过, 继续执行下一次循环。expression\_list 是被应用到每个集成员的表达式列表, 当用的是 @for 函数时, expression\_list 可以包含多个表达式, 其间用逗号隔开。这些表达式将被作为约束加到模型中。当使用其余的三个集循环函数时, expression\_list 只能有一个表达式。如果省略 set\_index\_list, 那么在 expression\_list 中引用的所有属性的类型都是 setname 集。

#### 1. @for

该函数用来产生对集成员的约束。基于建模语言的标量需要显式输入每个约束, 不过 @for 函数允许只输入一个约束, 然后 LINGO 自动产生每个集成员的约束。

#### 2. @sum

该函数返回遍历指定的集成员的一个表达式的和。

#### 3. @min 和 @max

返回指定的集成员的一个表达式的最小值或最大值。

### 5.8 输入和输出函数

输入和输出函数可以把模型和外部数据比如文本文件、数据库和电子表格等连接起来。

#### 1. @file 函数

该函数用从外部文件中输入数据, 可以放在模型中任何地方。该函数的语法格式为 @file('filename')。这里 filename 是文件名, 可以采用相对路径和绝对路径两种表示方式。@file 函数对同一文件的两种表示方式的处理和对两个不同的文件处理是一样的, 这一点必须注意。

当使用 @file 函数时, 可把记录的内容 (除了一些记录结束标记外) 看作是替代模型中 @file('filename') 位置的文本。这也就是说, 一条记录可以是声明的一部分, 整个声明, 或一系列声明。在数据文件中注释被忽略。注意在 LINGO 中不允许嵌套调用 @file 函数。

#### 2. @text 函数

该函数被用在数据部分用来把解输出至文本文件中。它可以输出集成员和集属性值。其语法为

@text(['filename'])

这里 filename 是文件名, 可以采用相对路径和绝对路径两种表示方式。如果忽略 filename, 那么数据就被输出到标准输出设备 (大多数情形都是屏幕)。@text 函数仅能出现在模型数据部分的一条语句的左边, 右边是集名 (用来输出该集的所有成员名) 或集属性名 (用来输出该集属性的值)。

把用接口函数产生输出的数据声明称为输出操作。输出操作仅当求解器求解完模型后才执行, 执行次序取决于其在模型中出现的先后。

#### 3. @ole 函数

@OLE 是从 EXCEL 中引入或输出数据的接口函数, 它是基于传输的 OLE 技术。OLE 传输直接在内存中传输数据, 并不借助于中间文件。当使用 @OLE 时, LINGO 先装载 EXCEL, 再通知 EXCEL 装载指定的电子数据表, 最后从电子数据表中获得 Ranges。为了使用 OLE 函数, 必须有 EXCEL5 及其以上版本。OLE 函数可在数据部分和初始部分引入数据。

@OLE 可以同时读集成员和集属性, 集成员最好用文本格式, 集属性最好用数值格式。原始集每个集成员需要一个单元(cell), 而对于 n 元的派生集每个集成员需要 n 个单元, 这

里第一行的  $n$  个单元对应派生集的第一个集成员，第二行的  $n$  个单元对应派生集的第二个集成员，依此类推。

@OLE 只能读一维或二维的 Ranges（在单个的 EXCEL 工作表(sheet)中），但不能读间断的或三维的 Ranges。Ranges 是自左而右、自上而下来读。

4. @ranged(variable\_or\_row\_name)

为了保持最优基不变，变量的费用系数或约束行的右端项允许减少的量。

5. @rangeu(variable\_or\_row\_name)

为了保持最优基不变，变量的费用系数或约束行的右端项允许增加的量。

6. @status()

返回 LINGO 求解模型结束后的状态：

0 Global Optimum（全局最优）

1 Infeasible（不可行）

2 Unbounded（无界）

3 Undetermined（不确定）

4 Feasible（可行）

5 Infeasible or Unbounded（通常需要关闭“预处理”选项后重新求解模型，以确定模型究竟是不可行还是无界）

6 Local Optimum（局部最优）

7 Locally Infeasible（局部不可行，尽管可行解可能存在，但是 LINGO 并没有找到一个）

8 Cutoff（目标函数的截断值被达到）

9 Numeric Error（求解器因在某约束中遇到无定义的算术运算而停止）

通常，如果返回值不是 0、4 或 6 时，那么解将不可信，几乎不能用。该函数仅被用在模型的数据部分来输出数据。

7. @dual

@dual(variable\_or\_row\_name)返回变量的判别数（检验数）或约束行的对偶（影子）价格（dual prices）。

## 5.9 辅助函数

1. @if(logical\_condition,true\_result,false\_result)

@if 函数将评价一个逻辑表达式 logical\_condition，如果为真，返回 true\_result，否则返回 false\_result。

2. @warn('text',logical\_condition)

如果逻辑条件 logical\_condition 为真，则产生一个内容为'text'的信息框。

## 六、LINGO WINDOWS 命令

### 6.1 文件菜单（File Menu）

1. 新建（New）

从文件菜单中选用“新建”命令、单击“新建”按钮或直接按 F2 键可以创建一个新的“Model”窗口。在这个新的“Model”窗口中能够输入所求解的模型。

2. 打开（Open）

从文件菜单中选用“打开”命令、单击“打开”按钮或直接按 F3 键可以打开一个已经存在的文本文件。这个文件可能是一个 Model 文件。

3. 保存(Save)

从文件菜单中选用“保存”命令、单击“保存”按钮或直接按 F4 键用来保存当前活动窗口

---

(最前台的窗口)中的模型结果、命令序列等保存为文件。

4. 另存为... (Save As... )

从文件菜单中选用“另存为...”命令或按 F5 键可以将当前活动窗口中的内容保存为文本文件,其文件名为你在“另存为...”对话框中输入的文件名。利用这种方法你可以将任何窗口的内容如模型、求解结果或命令保存为文件。

5. 关闭 (Close)

在文件菜单中选用“关闭”(Close)命令或按 F6 键将关闭当前活动窗口。如果这个窗口是新建窗口或已经改变了当前文件的内容, LINGO 系统将会提示是否想要保存改变后的内容。

6. 打印(Print)

在文件菜单中选用“打印” (Print)命令、单击“打印”按钮或直接按 F7 键可以将当前活动窗口中的内容发送到打印机。

7. 打印设置(Print Setup... )

在文件菜单中选用“打印设置...”命令或直接按 F8 键可以将文件输出到指定的打印机。

8. 打印预览(Print Preview)

在文件菜单中选用“打印预览...”命令或直接按 Shift+F8 键可以进行打印预览。

9. 输出到日志文件(Log Output... )

从文件菜单中选用“Log Output...”命令或按 F9 键打开一个对话框,用于生成一个日志文件,它存储接下来在“命令窗口”中输入的所有命令。

10. 提交 LINGO 命令脚本文件(Take Commands... )

从文件菜单中选用“Take Commands...”命令或直接按 F11 键就可以将 LINGO 命令脚本 (command script) 文件提交给系统进程来运行。

11. 引入 LINGO 文件(Import Lingo File... )

从文件菜单中选用“Import Lingo File...”命令或直接按 F12 键可以打开一个 LINGO 格式模型的文件,然后 LINGO 系统会尽可能把模型转化为 LINGO 语法允许的程序。

12. 退出 (Exit)

从文件菜单中选用“Exit”命令或直接按 F10 键可以退出 LINGO 系统。

## 6.2 编辑菜单(Edit Menu)

1. 恢复(Undo)

从编辑菜单中选用“恢复”(Undo)命令或按 Ctrl+Z 组合键,将撤销上次操作、恢复至其前的状态。

2. 剪切(Cut)

从编辑菜单中选用“剪切”(Cut)命令或按 Ctrl+X 组合键可以将当前选中的内容剪切至剪贴板中。

3. 复制(Copy)

从编辑菜单中选用“复制”(Copy)命令、单击“复制”按钮或按 Ctrl+C 组合键可以将当前选中的内容复制到剪贴板中。

4. 粘贴(Paste)

从编辑菜单中选用“粘贴”(Paste)命令、单击“粘贴”按钮或按 Ctrl+V 组合键可以将粘贴板中的当前内容复制到当前插入点的位置。

5. 粘贴特定.. (Paste Special..)

与上面的命令不同,它可以用于剪贴板中的内容不是文本的情形。

6. 全选(Select All)

从编辑菜单中选用“Select All”命令或按 Ctrl+A 组合键可选定当前窗口中的所有内容。

## 7. 匹配小括号(Match Parenthesis)

从编辑菜单中选用“Match Parenthesis”命令、单击“Match Parenthesis”按钮或按 Ctrl+P 组合键可以为当前选中的开括号查找匹配的闭括号。

## 8. 粘贴函数(Paste Function)

从编辑菜单中选用“Paste Function”命令可以将 LINGO 的内部函数粘贴到当前插入点。

# 6.3 LINGO 菜单

## 1. 求解模型 (Solve)

从 LINGO 菜单中选用“求解”命令、单击“Solve”按钮或按 Ctrl+S 组合键可以将当前模型送入内存求解。

## 2. 求解结果... (Solution...)

从 LINGO 菜单中选用“Solution...”命令、单击“Solution...”按钮或直接按 Ctrl+O 组合键可以打开求解结果的对话框。这里可以指定查看当前内存中求解结果的那些内容。

## 3. 查看... (Look...)

从 LINGO 菜单中选用“Look...”命令或直接按 Ctrl+L 组合键可以查看全部的或选中的模型文本内容。

## 4. 灵敏性分析 (Range, Ctrl+R)

用该命令产生当前模型的灵敏性分析报告:研究当目标函数的费用系数和约束右端项在什么范围(此时假定其它系数不变)时,最优基保持不变。灵敏性分析是在求解模型时作出的,因此在求解模型时灵敏性分析是激活状态,但是默认是不激活的。为了激活灵敏性分析,运行 LINGO|Options..., 选择 General Solver Tab, 在 Dual Computations 列表框中,选择 Prices and Ranges 选项。灵敏性分析耗费相当多的求解时间,因此当速度很关键时,就没有必要激活它。

## 5. 模型通常形式... (Generate...)

从 LINGO 菜单中选用“Generate...”命令或直接按 Ctrl+G 组合键可以创建当前模型的代数形式、LINGO 模型或 MPS 格式文本。

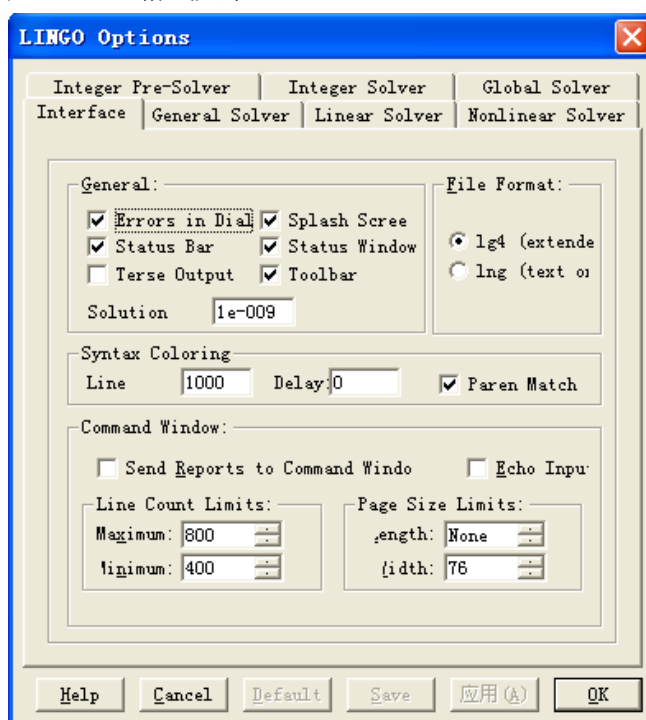


图 4 Lingo 系统参数选择对话框



## 6. 选项... (Options...)

从 LINGO 菜单中选用“Options...”命令、单击“Options...”按钮或直接按 Ctrl+I 组合键可以改变一些影响 LINGO 模型求解时的参数。该命令将打开一个含有 7 个选项卡的窗口，你可以通过它修改 LINGO 系统的各种参数和选项，如图 4 所示。

修改完以后，你如果单击“Apply (应用)”按钮，则新的设置马上生效；如果单击“OK (确定)”按钮，则新的设置马上生效，并且同时关闭该窗口。如果单击“Save (保存)”按钮，则将当前设置变为默认设置，下次启动 LINGO 时这些设置仍然有效。单击“Default (缺省值)”按钮，则恢复 LINGO 系统定义的原始默认设置 (缺省设置)。

### 6.4 窗口菜单 (Windows Menu)

#### 1. 命令行窗口 (Open Command Window)

从窗口菜单中选用“Open Command Window”命令或直接按 Ctrl+I 可以打开 LINGO 的命令行窗口。在命令行窗口中可以获得命令行界面，在“:”提示符后可以输入 LINGO 的命令行命令。

#### 2. 状态窗口 (Status Window)

从窗口菜单中选用“Status Window”命令或直接按 Ctrl+2 可以打开 LINGO 的求解状态窗口。

如果在编译期间没有表达错误，那么 LINGO 将调用适当的求解器来求解模型。当求解器开始运行时，它就会显示求解器状态窗口 (LINGO Solver Status)，见图 5。



图 5 Lingo 求解器状态窗

求解器状态窗口对于监视求解器的进展和模型大小是有用的。求解器状态窗口提供了一个中断求解器按钮 (Interrupt Solver)，点击它会导致 LINGO 在下一次迭代时停止求解。在绝大多数情况，LINGO 能够交还和报告到目前为止的最好解。一个例外是线性规划模型，返回的解是无意义的，应该被忽略。但这并不是一个问题，因为线性规划通常求解速度很快，很少需要中断。注意：在中断求解器后，必须小心解释当前解，因为这些解可能根本就不最优解、可能也不是可行解或者对线性规划模型来说就是无价值的。

在中断求解器按钮的右边的是关闭按钮 (Close)。点击它可以关闭求解器状态窗口，不

过可在任何时间通过选择 Windows|Status Window 再重新打开。

在 **中断求解器按钮** 的右边的是标记为 **更新时间间隔** (Update Interval) 的域。LINGO 将根据该域指示的时间 (以秒为单位) 为周期更新 **求解器状态窗口**。可以随意设置该域, 不过若设置为 0 将导致更长的求解时间——LINGO 花费在更新的时间会超过求解模型的时间。

**变量框 (Variables)**

Total 显示当前模型的全部变量数, Nonlinear 显示其中的非线性变量数, Integers 显示其中的整数变量数。非线性变量是指它至少处于某一个约束中的非线性关系中。

**约束 (Constraints) 框**

Total 显示当前模型扩展后的全部约束数, Nonlinear 显示其中的非线性约束数。非线性约束是该约束中至少有一个非线性变量。如果一个约束中的所有变量都是定值, 那么该约束就被剔除出模型 (该约束为真), 不计入约束总数中。

**非零 (Nonzeroes) 框**

Total 显示当前模型中全部非零系数的数目, Nonlinear 显示其中的非线性变量系数的数目。

**内存使用 (Generator Memory Used, 单位: K) 框**

显示当前模型在内存中使用的内存量。可以通过使用 LINGO|Options 命令修改模型的最大内存使用量。

**已运行时间 (Elapsed Runtime) 框**

显示求解模型到目前所用的时间, 它可能受到系统中别的应用程序的影响。

**求解器状态 (Solver Status) 框**

显示当前模型求解器的运行状态。

## 6.5 帮助菜单(Help Menu)

### 1. 帮助主题 (Help Menu)

从帮助菜单中选用“Help Menu”可以打开 LINGO 的帮助文件。

### 2. 关于 LINGO(About Lingo)

关于当前 LINGO 的版本信息等。

## 七、LINGO 的命令行命令

以下将按类型列出在 LINGO 命令行窗口中使用的命令, 每条命令后都附有简要的描述说明。

在平台中, 从的窗口菜单中选用“Command Window”命令或直接按 Ctrl+1 可以打开 LINGO 的命令行窗口, 便可以在命令提示符“:”后输入以下命令。

如果需要以下命令的详细描述说明, 可以查阅 LINGO 的帮助。

### 1. LINGO 信息

Cat        显示所有命令类型

Com        按类型显示所用 LINGO 命令

Help       显示所需命令的简要帮助信息

Mem        显示内存变量的信息

### 2. 输入(Input)

model      以命令行方式输入一个模型

take       执行一个文件的命令正本或从磁盘中读取某个模型文件

### 3. 显示(Display)

look       显示当前模型的内容

genl       产生 LINGO 兼容的模型

---

gen	生成并显示整个模型
hide	为模型设置密码保护
pause	暂停屏幕输出直至再次使用此命令
4. 文件输出(File Output)	
div	将模型结果输出到文件
svrt	将模型结果输出到屏幕
save	将当前模型保存到文件
smps	将当前模型保存为 MPS 文件
5. 求解模型(Solution)	
go	求解当前模型
solu	显示当前模型的求解结果
6. 编辑模型(Problem Editing)	
del	从当前模型中删除指定的某一行或某两行之间(包括这两行)的所有行
ext	在当前模型中添加几行
alt	用新字符串替换掉某一行中、或某两行之间的所有行中的旧字符串
7. 退出系统(Quit)	
quit	退出 LINGO 系统
8. 系统参数(System Parameters)	
page	以“行”为单位设置每页长度
ter	以简略方式输出结果
ver	以详细方式输出结果
wid	以“字符”为单位设置显示和输出宽度
set	重新设置默认参数
freeze	保存当前参数设置, 以备下一次重新启动 LINGO 系统时还是这样的设置
time	显示本次系统的运行时间

#### 参考文献

- [1]朱德通. 最优化模型与实验. 北京: 同济大学出版社, 2003.
- [2]谢金星、薛毅. 优化建模与 Lingo/Lingo 软件. 北京: 清华大学出版社, 2005.

## 附件3 时空网络

在为航空运输决策问题建模时，需要使用时空网络，现做些介绍。

### 一、时空网络的节点

在时空网络中，横轴方向给出航班飞行涉及到的机场，纵向为时间轴，每个机场一根时间轴，叫做时间线，从上到下标出每天第一个航班到最后一个航班之间的各时间点，时空网络的节点是该时间线上航班的出发时点或到达时点。图1中各时间线的圆圈点便是节点。可见节点是二维的：时间和机场（空间）。

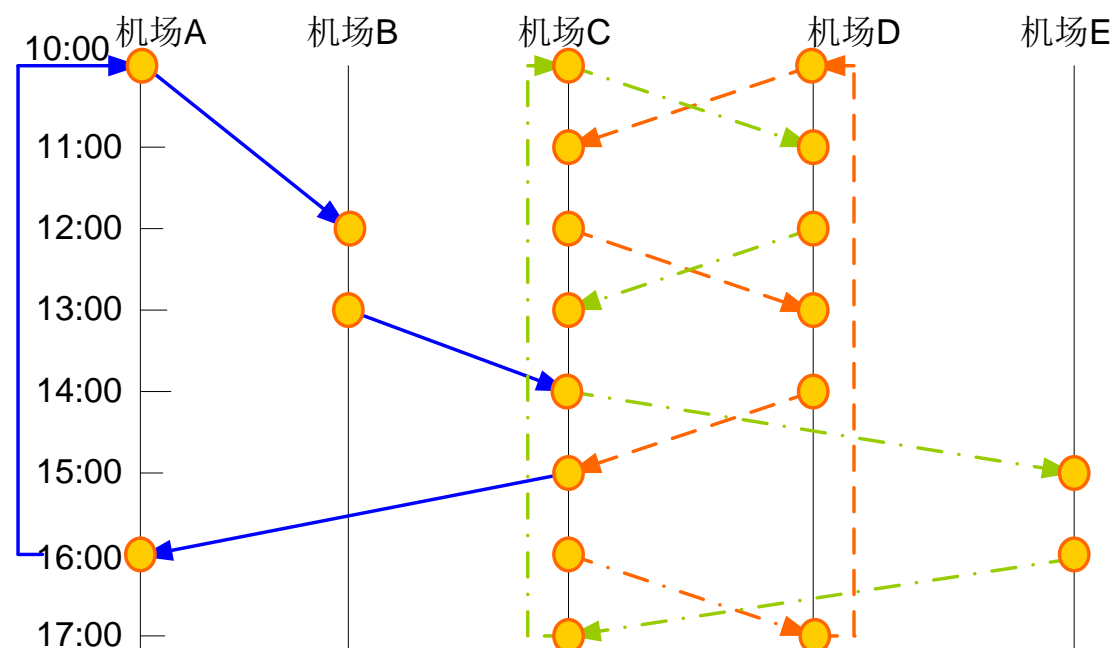


图1 时空网络

### 二、时空网络的边

在时空网络中，存在如下几种边：

#### 1、航班边

表示一个航班，从一个机场时间线的节点指向另一个机场时间的节点。这种边总是斜向下的。

#### 2、停车场边

表示停留在某一机场一段时间，从一个机场时间线的某节点指向同一机场时间线的另一节点。方向向下，由于方向确定，常常略去停车场边的方向。

#### 3、过夜边

表示某飞机在基地机场过夜，从该基地机场时间线的最后一个航班节点指向同一机场最早航班的节点。方向回溯向上，意思是说今天的飞行任务已经完成，等待明天的重新开始。

### 三、时空网络时间参数的计算

#### 1、航班轮挡时间

轮挡时间=该航班边的箭头节点的时间-箭尾节点的时间

#### 2、飞机停车场时间

停车场时间=该停车场边的箭头节点的时间-箭尾节点的时间

#### 四、时空网络建模

##### 1、节点飞机流平衡条件

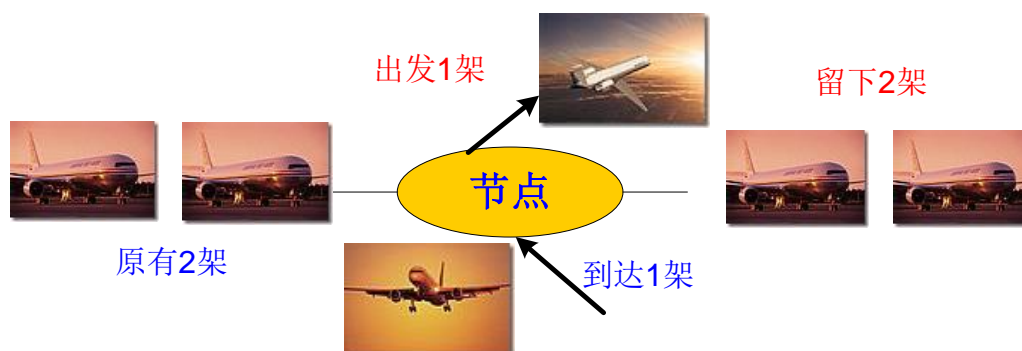


图2 节点处飞机流平衡条件

对于每种机型，在时空网络中各节点的飞机流必须保持平衡。图2中某机型有一个航班到达，一个航班出发，原有停车场飞机2架，因此该节点后，该机型仍有2架飞机停车场，即该机型留下飞机数=2（原有飞机数）+1（到达1架）-1（出发1架）。

例如对于本课程设计的机场L，它的时间线如图3所示。

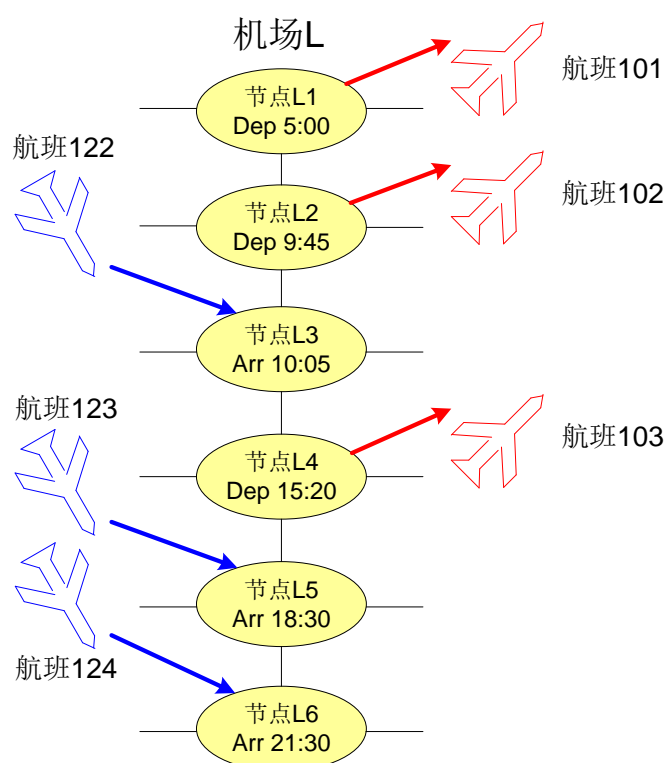


图3 机场L的时间线

则对于机型1有如下约束条件

$$G_{L1,1} = G_{L6,1} - x_{101,1}$$

$$G_{L2,1} = G_{L1,1} - x_{102,1}$$

$$G_{L3,1} = G_{L2,1} + x_{122,1}$$

$$G_{L4,1} = G_{L3,1} - x_{103,1}$$

$$G_{L5,1} = G_{L4,1} + x_{123,1}$$

$$G_{L6,1} = G_{L5,1} + x_{124,1}$$

---

## 2、飞机总数约束

每基地机场各机型过夜飞机数之和不超过该型飞机的总数。

对于本课程设计的机型 1 (Boeing737-800)，有如下的总数约束

$$\begin{aligned} &G_{L6,1} + G_{S6,1} + G_{B6,1} + G_{O6,1} + G_{A6,1} \\ &+ G_{I6,1} + G_{M6,1} + G_{J6,1} \leq 9 \end{aligned}$$

以上两种是主要约束条件，还可以根据实际问题的具体情况列出其他约束条件，这里不再赘叙，同学们可以在以后的工作中逐步掌握。

---

附件 4 设计报告模板

**《运筹学》课程设计报告**（黑体、二号、居中）

姓名：\_\_\_\_\_（黑体、四号）

班级：\_\_\_\_\_

学号：\_\_\_\_\_

---

## 一、问题描述（一级标题黑体、小四）

机型指派问题...（宋体、五号）

### 1. 二级标题（黑体、五号）

机型指派问题...（宋体、五号）

#### （1）三级标题（黑体、五号）

机型指派问题...（宋体、五号）

## 二、分析建模

同上

## 三、模型求解

同上

## 四、结果分析

同上

## 五、结论

同上

（课程设计正文 1.2 倍行距）



参考文献：

[1]朱德通. 最优化模型与实验. 北京：同济大学出版社，2003

[2]...

（参考文献：宋体、五号、1.2 倍行距）

