第 11 讲　MATLAB CUMCM 真题求解实例二：优化型

作者：卓金武，MathWorks 中国

2003 年的 B 题是典型的优化型问题，这道问题的特色是模型容易建立，但求解比较困难。这道题目在求解方面的难点是模型有交叉，所以当时我们的求解策略是分步求解、逐级优化，采用这种策略后，就可以将复杂的优化问题转化为标准的规划模型进行求解了。

在 2003 年 MATLAB 还没有整数规划求解器，在这一讲就介绍用 MATLAB 的整数规划求解器来求解这一问题。

1 问题的提出

钢铁工业是国家工业的基础之一，铁矿是钢铁工业的主要原料基地。许多现代化铁矿是露天开采的，它的生产主要是由电动铲车（以下简称电铲）装车、电动轮自卸卡车（以下简称卡车）运输来完成。提高这些大型设备的利用率是增加露天矿经济效益的首要任务。

露天矿里有若干个爆破生成的石料堆，每堆称为一个铲位，每个铲位已预先根据铁含量将石料分成矿石和岩石。一般来说，平均铁含量不低于 25%的为矿石，否则为岩石。每个铲位的矿石、岩石数量，以及矿石的平均铁含量（称为品位）都是已知的。每个铲位至多能安置一台电铲，电铲的平均装车时间为 5 分钟。

卸货地点（以下简称卸点）有卸矿石的矿石漏、2 个铁路倒装场（以下简称倒装场）和卸岩石的岩石漏、岩场等，每个卸点都有各自的产量要求。从保护国家资源的角度及矿山的经济效益考虑，应该尽量把矿石按矿石卸点需要的铁含量（假设要求都为 29.5% $\pm$ 1%，称为品位限制）搭配起来送到卸点，搭配的量在一个班次（8 小时）内满足品位限制即可。从长远看，卸点可以移动，但一个班次内不变。卡车的平均卸车时间为 3 分钟。

所用卡车载重量为 154 吨，平均时速 28 km/h 。卡车的耗油量很大，每个班次每台车消耗近 1 吨柴油。发动机点火时需要消耗相当多的电瓶能量，故一个班次中只在开始工作时点火一次。卡车在等待时所耗费的能量也是相当可观的，原则上在安排时不应发生卡车等待的情况。电铲和卸点都不能同时为两辆及两辆以上卡车服务。卡车每次都是满载运输。

每个铲位到每个卸点的道路都是专用的宽 60 m 的双向车道，不会出现堵车现象，每段道路的里程都是已知的。

一个班次的生产计划应该包含以下内容：出动几台电铲，分别在哪些铲位上；出动几辆卡车，分别在哪些路线上各运输多少次（因为随机因素影响，装卸时间与运输时间都不精确，所以排时计划无效，只求出各条路线上的卡车数及安排即可）。一个合格的计划要在卡车不等待条件下满足产量和质量（品位）要求，而一个好的计划还应该考虑下面两条原则之一：

1.总运量（吨公里）最小，同时出动最少的卡车，从而运输成本最小；

2.利用现有车辆运输，获得最大的产量（岩石产量优先；在产量相同的情况下，取总运量最小的解）。

请你就两条原则分别建立数学模型，并给出一个班次生产计划的快速算法。针对下面的实例，给出具体的生产计划、相应的总运量及岩石和矿石产量。

某露天矿有铲位 10 个，卸点 5 个，现有铲车 7 台，卡车 20 辆。各卸点一个班次的产量要求：矿石漏 1.2 万吨、倒装场 I 1.3 万吨、倒装场 II 1.3 万吨、岩石漏 1.9 万吨、岩场 1.3

万吨。

铲位和卸点位置的二维示意图如图 1 所示，具体的数据参考官网的赛题。

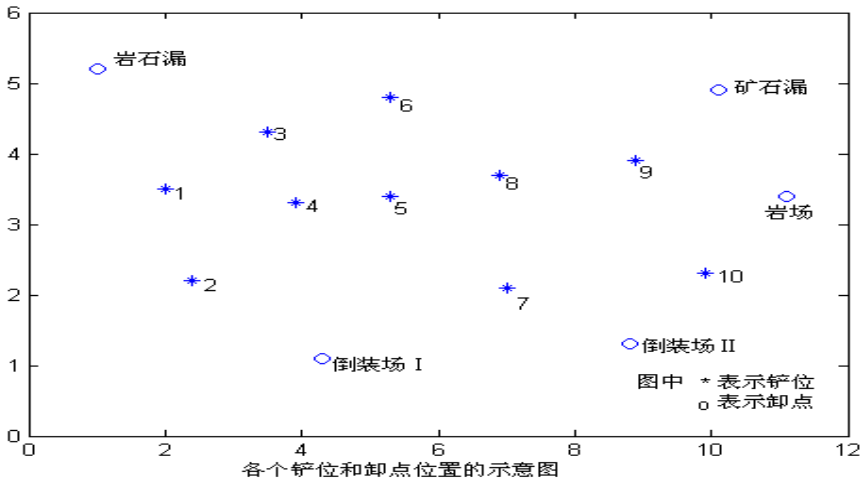

图11-1　矿场铲位和卸点示意图

2 基本假设

为了便于问题的研究，我们对题目中的不确定因素作一些约定和假设。

1) 电铲在一个班次内不改变铲位，也就是说每台电铲在一个班次内只在一个铲位上工作。这主要是因为电铲的转移不方便并且电铲的转移需要占用时间，影响公司的效益。

2) 矿石漏和铁路倒装场只是卸矿石的不同地方，它们的开采对露天矿的经济效益无影响。同样，卸岩石的岩石漏和岩场的属性也不影响开采公司的经济效益。开采公司的经济效益主要与开采量与运输成本有关。

3) 卸点的品位是指在一个班次内在卸点内所卸总矿石铁的综合含量 ，并不要求任何一部分矿石的铁含量达到品位限制要求。

4) 卡车每次运输都按载重量满载运输，并不考虑因颠簸而使岩石或矿石减少的情况。另外卡车运输始终以 28km/h 的平均速度行驶，发动和刹车所占用的时间忽略不计。

5) 在同一班次内，每辆卡车所走的路线是不定的，即卡车选择哪条路线是随机的。

3 问题的分析及模型的准备

通过直观的分析可知，本问题是一个较复杂的运输系统调度问题。问题要求分别满足两条运输原则的条件下建立一个班次运输方案安排的数学模型，并且要给出所用电铲的台数，每台电铲的铲位，出动卡车的数量，卡车的具体调度安排等。所以本问题是一个大型的目标规划问题，目标函数是要求的两个原则，即一个是要求总运量最小，同时出动的卡车最少，另一个是要求获得最大的产量。对于开采公司来说，制定的两个原则事实上就是减少成本，增加收入，以提高公司的经济效益。这样我们就可知道研究该问题的方向就是找目标函数，抽象约束条件，建立规划问题的数学模型。

为了建立完善的数学模型，我们还需对问题作进一步的分析。

(1) 运输矩阵的建立

卡车运输路线的选择是双向、随机的，当多辆卡车同时运输时，他们所形成的运输网错综复杂。为了便于描述卡车在一个班次的调动状态，我们先规定了两个运输方向，同人们习惯上的方向相同，我们把从铲位到卸点的方向称为前进（Go）方向，而将从卸点到铲位的方向称为返回(Return)方向。

设有 m 个卸点，n 个铲位，可以构建以下矩阵描述 Go 方向的运输状态：

$$G=\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{mn} \end{bmatrix}$$

其中，g_{ij} 表示在一个班次内从 j 铲位到 i 卸点单向路径上所通过的总车次，$1\le j\le n,1\le i\le m$。

同理，我们可得到 Return 矩阵：

$$R=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{mn} \end{bmatrix}$$

其中 r_{ij} 表示在一个班次内从 i 卸点到 j 铲位单向路径上所通过的总车次。

我们将 Go 矩阵和 Return 矩阵统称为调度矩阵。

(2) 原则 1 的数学分析

原则 1 要求总运量最小，同时出动的卡车数量最少，这实际上是要求运输成本最小，所以原则 1 又可称为成本最小原则。这里的总运量我们理解为卡车所装载的货物总质量（吨）与卡车在装载状态下所行的路程之积，其数字表达式为

$$Q=\sum_{i=1}^{m}\sum_{j=1}^{n}ag_{ij}s_{ij}$$

其中 s_{ij} 表示从铲位 j 到卸点 i 之间的路程，a 为卡车满载时的载重。

当卡车从卸点返回时，此时虽然卡车所走的路程不为零，但此时卡车所装载货物的质量为零，所以返回时卡车的运量为零，所以卡车的总运输指的是从铲位到卸点也即 Go 方向上的总运量。

原则 1 同时也要求在同一班次内出动卡车的数量最少。卡车最少的运输状态有以下两个特点：

卡车得到最大限度的利用，即卡车几乎没有等待时间（闲置时间）。

卡车充分地工作，恰能完成运输问题，或者超额的部分并不多。

对于多辆卡车的装、运、卸的时间我们很难确定，但根据特点（1），我们在宏观上很容易找到卡车数量与其他因素之间的关系。

由于所有卡车几乎一直在工作，即对每辆卡车来说在一个班次内都处于装、运、卸三个时间状态，所以我们将所有卡车的工作时拆合成一辆卡车的工作时，便有

$$NT=\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\left(g_{ij}+r_{ij}\right)s_{ij}/v\right]+\sum_{i=1}^{m}\sum_{j=1}^{n}\left(g_{ij}t_{上}+r_{ij}t_{下}\right)+t^{*}$$

其中 T 为生产周期，即一个班次的时间，t^* 为在一个班次内所有卡车的总等待时间，于是有

$$N=\left\{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\left(g_{ij}+r_{ij}\right)s_{ij}/v\right]+\sum_{i=1}^{m}\sum_{j=1}^{n}\left(g_{ij}t_{上}+r_{ij}t_{下}\right)+t^*\right\}/T$$

由于整个运输过程中原则上不应存在等待时间，所以 t^* 的值应近似为零或就是零。

(3) 原则 2 的数学分析

原则 2 要求利用现有车辆,获得最大的产量,所以原则 2 又可称为产量最大原则,这里的产量指的是矿石和岩石的总产量，其数学表达式为：

$$F=\sum_{i=1}^{m}\sum_{j=1}^{n}ag_{ij}$$

(4) 等待时间的控制

我们在安排运输方案时,原则上不应存在等待时间,但不排除一定存在等待时间的情况,所以我们安排运输时应尽可能避免出现等待时间的情况。根据参考文献[8]，卡车在进行调度时可以根据“最小饱和度”调度准则（MSD），以尽可能地避免发生等待现象。

这一准则的实质，是将卡车调往具有最小“饱和”程度的路线：

$$\begin{cases}choice(i)=j\Big|_{1\le j\le n}^{\min\{D_j\}}\\ choice(j)=i\Big|_{1\le i\le m}^{\min\{D_i\}}\end{cases}$$

式中 choice(i)表示处于 i 卸点的待发车所选择的将去铲位的代号，choice(j) 表示处于 j 铲位的待发车所选择的将去卸点的代号，D_j 表示由卸点到 j 铲位的饱和度，D_i 表示由铲位到 i 卸点的饱和度。D_j 和 D_i 的具体表达式为：

$$\begin{cases}D_j=\dfrac{s_{ij}\left(\dfrac{t'}{t_{上}}+N_j\right)}{vt_{上}}\\ D_i=\dfrac{s_{ij}\left(\dfrac{t'}{t_{下}}+N_i\right)}{vt_{下}}\end{cases}$$

其中 t' 为正装车（卸车）估计剩余的装车（卸车）时间，N_j 表示到第 j 号铲位的卡车数，不包括正装的卡车，N_i 表示到 i 卸点的卡车数，不包括正卸的卡车。

4 原则 1 数学模型（模型 1）的建立与求解

4.1 模型的建立

由上面问题的分析，我们给出了成本的数学表达式，再经过对目标函数约束条件的分析后，建立了以下的双目标线性规划模型：

Goals:

$$\min Q = \sum_{i=1}^{m}\sum_{j=1}^{n} a g_{ij} s_{ij}$$

$$\min N = \left\{ \sum_{i=1}^{m}\sum_{j=1}^{n} (g_{ij} + \gamma_{ij}) s_{ij} / v + \sum_{i=1}^{m}\sum_{j=1}^{n} (g_{ij} t_{上} + \gamma_{ij} t_{下}) + t^{*} \right\} / T$$

S.T:

$$\sum_{j=1}^{n} a g_{ij} \geq K_i \tag{1}$$

$$\sum_{j=1}^{n} b_j g_{ij} a / \sum_{j=1}^{n} g_{ij} a \in [29.5\% \pm 1\%] \tag{2}$$

$$\sum_{i=1}^{m} a g_{ij} \leq M_j \tag{3}$$

$$\sum_{i=1}^{m} a g_{ij} \leq U_j \tag{4}$$

$$\sum_{i=1}^{m} g_{ij} \leq \frac{T}{t_{上}}, 1 \leq j \leq n \tag{5}$$

$$\sum_{j=1}^{n} r_{ij} \leq \frac{T}{t_{下}}, 1 \leq i \leq m \tag{6}$$

$$T^{*} \geq 0, \lim T^{*} = 0 \tag{7}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \tag{8}$$

$$g_{ij} \geq 0, r_{ij} \geq 0, 且 g_{ij}, r_{ij} 为整数 \tag{9}$$

$$choice(i) = j \left| \begin{matrix} \min\{D_j\} \\ 1 \leq j \leq n \end{matrix} \right. \tag{10}$$

$$choice(j)=i\left|\begin{matrix}\min\{D_i\}\\1\le i\le m\end{matrix}\right. \tag{11}$$

关于约束条件的说明

1） 约束条件（1）是为保障在一个班次内要满足各卸点的需求；

2） 条件（2）是对铲位搭配的约束，即在同一班次内所有矿石的卸点都要达到品位要求的限制；

3） 条件（3）（4）是基于铲位的岩石和矿石的储量都是有限的而进行的约束，即从任何铲位所输出的产量不应超过该铲位的储量；

4） 条件（5）（6）是对g_{ij}和r_{ij}的约束，它们的上限不应超过$T/t_{上}$，$T/t_{下}$；

5） 条件（7）描述了等待时间的情形，说明了可以存在等待时间，但尽量应使等待时间为0；

6） 条件（8）给出了 Go 和 Return 矩阵元素之间的逻辑关系；

7） 条件（9）是对目标函数中g_{ij}和r_{ij}的约束，这是由它们的现实意义而定的。

8） 条件（10）和条件（11）是为了保证尽量避免等待现象而进行的实时调度的约束。

4.2 模型的求解

模型 1 是典型的大型的双目标线性规划问题，即使在约束条件下对两个目标分别求解，也是困难的，困难在于模型中的变量太多，尤其是模型的约束条件中包含了实时调度的限制，这种限制使模型变成非线性，而且不易控制的复杂的数学模型。因此不易直接由计算机进行搜索求解，只能另辟途径。

（1）模型算法的理论分析

模型的求解要求给出一个班次内出动电铲的台数、电铲分布的铲位、出动卡车的数量及卡车的路线分配。模型的目标函数为总运量最小，同时要求出动的卡车也是最少，但也要满足运输要求，所以我们先不考虑出动卡车的台数，直接以总运量最小为目标，求解模型。求解出运输方案后，卡车数量即可给出。

直接的求解很复杂，为此我们采取分步求解的方法：

第一步：用线性规划的方法求出从每个铲位到每个卸点所发的车次，从而求解出 Go 矩阵。

第二步：从 Go 矩阵判断所需出动的电铲的台数和铲位分配。

第三步：依据 Go 矩阵提供的信息，用线性规划方法求出由每个卸点返回到每个铲位的车次，从而给出了 Return 矩阵。

第四步：依据 Go 矩阵和 Return 矩阵，根据卡车的充分利用条件求出在一个班次内所需卡车的数量。

（2）分步求解的实现

为求解 Go 矩阵，我们先要求出从每个铲位到每个卸点的岩石或矿石的运量。为此，我们以总运量最小为目标函数，供应约束、需求约束、品位限制为约束条件，建立如下单目标线性规划数学模型。

$$\min Q=a\sum_{i=1}^{5}\sum_{j=1}^{10}g_{ij}s_{ij}$$

S.T:

$$\sum_{j=1}^{10} ag_{ij} \geq K_i$$

$$\sum_{i=1}^{5} ag_{ij} \leq M_j$$

$$\sum_{i=1}^{5} ag_{ij} \leq U_j$$

$$\sum_{j=1}^{5} b_j g_{ij} a / \sum_{j=1}^{5} g_{ij} a \in [29.5\% \pm 1\%]$$

$$\sum_{i=1}^{m} g_{ij} \leq \frac{T}{t_{上}}, 1 \leq j \leq n$$

$$g_{ij} \geq 0, \text{且为整数}$$

$$\sum_{j=1}^{10} g_{ij} \leq 160, \sum_{i=1}^{5} g_{ij} \leq 96$$

该模型是线性规问题，用 MATLAB 的求解程序如下：

```
%% 2003-B MATLAB 整数规划求解程序
c=[5.26    5.19   4.21   4      2.95   2.74   2.46   1.9
    0.64   1.27   1.9    0.99   1.9    1.13   1.27   2.25
    1.48   2.04   3.09   3.51   4.42   3.86   3.72   3.16
    2.25   2.81   0.78   1.62   1.27   0.5    5.89   5.61
    5.61   4.56   3.51   3.65   2.46   2.46   1.06   0.57
    0.64   1.76   1.27   1.83   2.74   2.6    4.21   3.72
    5.05   6.1];
A=[0.0154  0.0154 0.0154 0.0154 0.0154 0.0154 0.0154 0.0154
    0.0154
    0.0154.....................................................
.............................]; % 省略部分数据
b=[1.2154  -1.2   1.3154 -1.3   1.3154 -1.3   0      0      0
    0      0      0      61.68831169   68.18181818
    64.93506494   68.18181818   71.42857143   81.16883117
    68.18181818   84.41558442   87.66233766   81.16883117
    81.16883117   71.42857143   87.66233766   68.18181818
    74.67532468   87.66233766   68.18181818   74.67532468
    87.66233766   81.16883117   160    160    160    160
```

```
    160    96    96    96    96    96    96    96
    96     96    96]';
Aeq=[];
beq=[];
n = size(c,2);
lb=zeros(1,n);
ub=ones(1,n)*inf;
cons=1:n;
x=intlinprog(c,cons,A,b,Aeq,beq,lb,ub);

%% 解的转化

X = reshape(x,[5, 10])

%% 数据和模型可以参考《MATLAB 在数学建模中的应用》(第二版)
```

执行该程序求出一组解，由于该函数求出的解并非整数，所以我们用手工改动的方法对求出的结果进行优化处理，处理原则是对所得结果进行向上或向下取整，并在满足限制条件下使目标函数尽可能的大，这样我们便得到 Go 矩阵：

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	13.0000	0	40.0000	0	15.0000	2.0000	0	0	0	0
3	0	52.0000	0	0	0	0	0	0	0	0
4	0	0	45.0000	0	0	0	0	0	0	0
5	0	13.0000	0	0	0	68	0	0	0	0

用同样的程序，只要更新其中的变量，就可以轻松得到 Return 矩阵，这样就完成了整个模型的求解了，是不是发现 MATLAB 整数规划的求解也是挺方便的。

5 点评

优化问题的建模思路就是：定目标-抽约束-求解三步曲，该模型的建模过程相对来说比较容易些。但本问题的求解是比较麻烦的，虽然目标和约束非常清晰。从本篇的建模和求解过程来看，可以得到以下的技巧：

(1) 在建立模型的过程中，可以有个循序渐进的过程，从简单到复杂，这样求解的起点就会低些，更容易得到简单模型的解，这样求解也会有个逐渐深入的过程，也便于对模型的结果进行比较。

(2) 文中的模型，直接进行求解是非常困难的，这里面采取了分步求解、逐级优化的策略，将复杂的模型简化到了可以求解的程度，从而很容易对模型进行了求解。这样虽然得到的解不是最优解，但应该也是非常接近最优解的可行解。得到一个近似最优解，总比得不到解好吧，所以对模型进行简化在实际的建模比较中也要考虑到。

关于作者

卓金武，MathWorks 中国高级工程师，教育业务经理，在数据分析、数据挖掘、机器学习、数学建模、量

化投资和优化等科学计算方面有多年工作经验，现主要负责 MATLAB 校园版业务；曾 2 次获全国大学生数学建模竞赛一等奖, 1 次获全国研究生数学建模竞赛一等奖 ;专著 3 部：《MATLAB 在数学建模中的应用》、《大数据挖掘：系统方法与实例分析》、《量化投资：MATLAB 数据挖掘技术与实践》。