

# 多商品流问题及数值实验

张晋

北京航空航天大学，数学与系统科学学院

2017 年 10 月 8 日

## 1 引言

多商品流问题 (MultiCommodity Flow Problem, MCFP) 是多种物品在网络中从不同的源点流向不同的汇点的网络流问题。

## 2 多商品网络流模型

已知一有向网络  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , 其中  $\mathcal{N}$  是节点集,  $\mathcal{E}$  是弧集. 对每条弧  $l = (i, j) \in \mathcal{E}$ , 设其容量是  $c_l$ , 代表每条弧上所能承受负载的上界.

已知有  $M$  件物品  $K_1, K_2, \dots, K_M$ , 定义为  $K_m = (s_m, t_m, d_m)$ , 其中  $s_m$  和  $t_m$  是物品  $m$  的源点及汇点,  $d_m$  是流量需求, 这代表流量在节点  $s_m$  流入网络, 然后在节点  $t_m$  流出网络,  $d_m$  代表物品  $m$  流出减流入的量。

我们设物品  $m$  沿弧  $l$  的流量是  $f_{ml}$ , 则该流量分配需要满足以下条件:

a) 容量约束:

$$f_l = \sum_{m=1}^M f_{ml} \leq c_l \quad \forall l \in \mathcal{E} \quad (1)$$

b) 流平衡约束:

$$\sum_{l:l=(i,j) \in \mathcal{E}} f_{ml} - \sum_{l:l=(j,i) \in \mathcal{E}} f_{ml} = b_{mi} = \begin{cases} 0, & i \neq s_m, t_m \\ d_m, & i = s_m \\ -d_m, & i = t_m \end{cases} \quad (2)$$

c) 流非负:

$$f_{ml} \geq 0, \quad m = 1, \dots, M, \forall l \in \mathcal{E} \quad (3)$$

令  $\mathbf{A}$  是网络  $\mathcal{G}$  的点弧关联矩阵, 即  $N \times E$  阶矩阵, 且第  $l$  列与弧  $l = (i, j)$  对应, 仅第  $i$  行的元素为 1, 第  $j$  行的元素为 -1, 其余元素为 0. 再令  $\mathbf{b}_m = (b_{m1}, \dots, b_{mN})^T$ ,  $\mathbf{f}_m = (f_{m1}, f_{m2}, \dots, f_{mE})^T$ , 则可将等式约束 (2) 表示成

$$\mathbf{A} \mathbf{f}_m = \mathbf{b}_m$$

### 3 最优流量工程

若存在流分布  $\mathbf{f}$  满足流量约束, 我们应该思考怎样改变  $\mathbf{f}$  使得在其满足流量约束条件的同时, 每条弧的利用率都不高, 也就是说尽量避免某条弧堵塞的情况。

对此我们可以给定一个成本函数  $\phi(\mathbf{f})$  来评价整体网络的堵塞情况, 这里  $\phi(\mathbf{f})$  是  $\mathbf{f}$  的非减函数。最优流量工程即意味着在满足多商品流约束条件下, 使得成本函数  $\phi(\mathbf{f})$  取得极小值, 那么多商品流问题可转化为解决一个线性规划问题:

$$\min \phi(\mathbf{f}, \mathbf{c}) \quad (4)$$

$$\text{s.t. } \mathbf{f} = \sum_{m=1}^M \mathbf{f}_m \quad (5)$$

$$\mathbf{A}\mathbf{f}_m = \mathbf{b}_m, \quad m = 1, \dots, M \quad (6)$$

$$\mathbf{f}_m \geq \mathbf{0}, \quad m = 1, \dots, M \quad (7)$$

在所有关于流量工程的论文中, 最大弧利用率 (Maximum Arc Utilization, MAU) 和 M/M/1 延迟公式的逐段线性近似是两个使用最多的成本函数

极小化 MAU 确保最拥塞的热点弧利用率尽可能的小, 这里的 MLU 可以用弧上的负载和容量表述为

$$\phi(\mathbf{f}) = \max_{l \in \mathcal{E}} f_l / c_l \quad (8)$$

基于该目标函数的多商品流问题可转化为以下线性规划:

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to } \sum_{m=1}^M f_{ml} / c_l - z \leq 0, \quad \forall l \in \mathcal{E} \\ & \quad \mathbf{A}\mathbf{f}_m = \mathbf{b}_m, \quad m = 1, \dots, M \\ & \quad \mathbf{f}_m \geq \mathbf{0}, \quad m = 1, \dots, M \end{aligned} \quad (9)$$

M/M/1 延迟公式的逐段线性近似由 Fortz 等提出, 是经作者与贝尔实验室的技术人员讨论后得到的. 下面将它简称它为 FT 成本函数, 可以将 FT 成本函数表述为  $\phi(\mathbf{f}) = \sum_{l \in \mathcal{E}} \phi(f_l)$ , 其中

$$\phi(f_l) = \begin{cases} f_l, & \frac{f_l}{c_l} \leq \frac{1}{3} \\ 3f_l - \frac{2}{3}c_l, & \frac{1}{3} < \frac{f_l}{c_l} \leq \frac{2}{3} \\ 10f_l - \frac{16}{3}c_l, & \frac{2}{3} < \frac{f_l}{c_l} \leq \frac{9}{10} \\ 70f_l - \frac{178}{3}c_l, & \frac{9}{10} < \frac{f_l}{c_l} \leq 1 \\ 500f_l - \frac{1468}{3}c_l, & 1 < \frac{f_l}{c_l} \leq \frac{11}{10} \\ 5000f_l - \frac{16318}{3}c_l, & \frac{11}{10} < \frac{f_l}{c_l} < \infty \end{cases} \quad (10)$$

基于 FT 成本函数的多商品流问题也可转化为线性规划问题:

$$\begin{aligned} & \text{minimize} && \sum_{l \in \mathcal{E}} z_l \\ & \text{subject to} && \sum_m f_{ml} - z_l \leq 0, && \forall l \in \mathcal{E} \\ & && 3 \sum_m f_{ml} - z_l \leq \frac{2}{3}c_l, && \forall l \in \mathcal{E} \\ & && 10 \sum_m f_{ml} - z_l \leq \frac{16}{3}c_l, && \forall l \in \mathcal{E} \\ & && 70 \sum_m f_{ml} - z_l \leq \frac{178}{3}c_l, && \forall l \in \mathcal{E} \\ & && 500 \sum_m f_{ml} - z_l \leq \frac{1468}{3}c_l, && \forall l \in \mathcal{E} \\ & && 5000 \sum_m f_{ml} - z_l \leq \frac{16318}{3}c_l, && \forall l \in \mathcal{E} \\ & && \mathbf{A}\mathbf{f}_m = \mathbf{b}_m, && m = 1, \dots, M \\ & && \mathbf{f}_m \geq \mathbf{0}, && m = 1, \dots, M \end{aligned} \quad (11)$$

## 4 数值实验

我们先来看 TE 中的一个经典例子，这个网络有 7 个节点 13 条弧，每条弧的容量是 5 个单位。此外有四个需求量均为 4 个单位的源-目的对 ( $M = 4$ )，具体的源节点、目的节点信息如图 1 所示。表 1 为点弧关联矩阵。这里为了简单，省去了未用到的弧。此外，弧上的数字表示弧的编号，此时  $\mathbf{c}^T = (5, 5, \dots, 5)_{1 \times 13}$

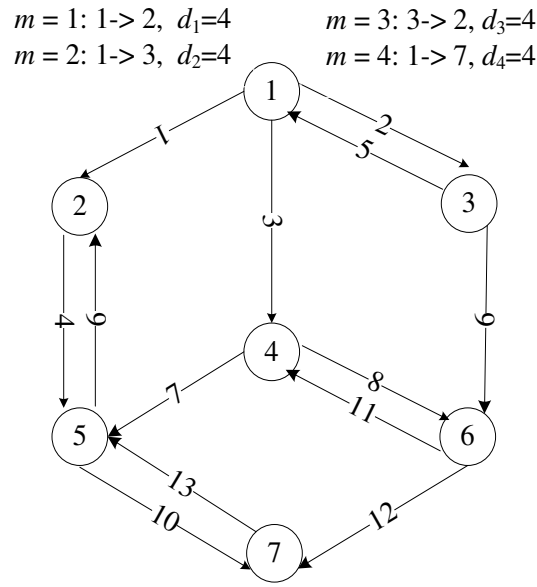


图 1: 经典例子

$$\mathbf{b}_1 = \begin{bmatrix} 4 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 4 \\ 0 \\ -4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 0 \\ -4 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_4 = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4 \end{bmatrix}$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 1 & 1 & & -1 & & & & & & \\ -1 & & & & 1 & & & & -1 & & \\ & -1 & & & 1 & 1 & & & & & \\ & & -1 & & & & 1 & 1 & & & -1 \\ & & & -1 & & & -1 & & 1 & 1 & \\ & & & & -1 & & -1 & & & & -1 \\ & & & & & -1 & & -1 & & 1 & 1 \\ & & & & & & & & -1 & & -1 & 1 \end{bmatrix}$$

表 1: 点弧关联矩阵

在此我们对该例子进行数值实验，考虑到 MATLAB 线性规划函数 `linprog` 的自变量只能为向量，我们需要对原线性规划进行变形，变成如下形式：

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}^T \mathbf{x} \\ & \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \\ & \mathbf{A}_{eq} \cdot \mathbf{x} = \mathbf{b}_{eq} \\ & \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{aligned} \tag{12}$$

## 4.1 MAU

在使用 MAU 方法时, 需进行以下变形:

$$\mathbf{x} = \begin{bmatrix} \mathbf{f}_{1l} \\ \mathbf{f}_{2l} \\ \mathbf{f}_{3l} \\ \mathbf{f}_{4l} \\ z \end{bmatrix}_{53 \times 1}, \quad \mathbf{f}_{il} = \begin{bmatrix} f_{i1} \\ f_{i2} \\ f_{i3} \\ \vdots \\ f_{i13} \end{bmatrix}_{13 \times 1}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{53 \times 1}, \quad \mathbf{f}^T \mathbf{x} = z$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} & -\mathbf{c} \end{bmatrix}_{13 \times 53}, \quad \mathbf{A} \cdot \mathbf{x} = \mathbf{f}_{1l} + \mathbf{f}_{2l} + \mathbf{f}_{3l} + \mathbf{f}_{4l} - z \cdot \mathbf{c}$$

其中  $\mathbf{I}$  是  $13 \times 13$  的单位向量,  $\mathbf{b}$  为  $13 \times 1$  的零向量.

$$\mathbf{Aeq} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_1 & & \mathbf{0} \\ & & \mathbf{A}_1 & \mathbf{0} \\ & & & \mathbf{A}_1 & \mathbf{0} \end{bmatrix}_{28 \times 53}$$

$$\mathbf{Aeq} \cdot \mathbf{x} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_1 & & \mathbf{0} \\ & & \mathbf{A}_1 & \mathbf{0} \\ & & & \mathbf{A}_1 & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{f}_{1l} \\ \mathbf{f}_{2l} \\ \mathbf{f}_{3l} \\ \mathbf{f}_{4l} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \mathbf{f}_{1l} \\ \mathbf{A}_1 \mathbf{f}_{2l} \\ \mathbf{A}_1 \mathbf{f}_{3l} \\ \mathbf{A}_1 \mathbf{f}_{4l} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{bmatrix}_{28 \times 1} = \mathbf{beq}$$

$\mathbf{lb}$  为  $53 \times 1$  的零向量,  $\mathbf{ub}$  为空.

## 4.2 M/M/1

在对 M/M/1 延迟公式进行计算时, 需要进行以下变形:

$$\mathbf{x} = \begin{bmatrix} f_{1l} \\ f_{2l} \\ f_{3l} \\ f_{4l} \\ z_l \end{bmatrix}_{65 \times 1}, \quad \mathbf{z}_l = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{13} \end{bmatrix}_{13 \times 1}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}_{(52+13) \times 1}, \quad \mathbf{f}^T \mathbf{x} = \sum_{l \in \mathcal{E}} z_l$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}, & \mathbf{I}, & \mathbf{I}, & \mathbf{I}, & -\mathbf{I} \\ 3\mathbf{I}, & 3\mathbf{I}, & 3\mathbf{I}, & 3\mathbf{I}, & -\mathbf{I} \\ 10\mathbf{I}, & 10\mathbf{I}, & 10\mathbf{I}, & 10\mathbf{I}, & -\mathbf{I} \\ 70\mathbf{I}, & 70\mathbf{I}, & 70\mathbf{I}, & 70\mathbf{I}, & -\mathbf{I} \\ 500\mathbf{I}, & 500\mathbf{I}, & 500\mathbf{I}, & 500\mathbf{I}, & -\mathbf{I} \\ 5000\mathbf{I}, & 5000\mathbf{I}, & 5000\mathbf{I}, & 5000\mathbf{I}, & -\mathbf{I} \end{bmatrix}_{78 \times 65}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \frac{2}{3}\mathbf{c}_l \\ \frac{16}{3}\mathbf{c}_l \\ \frac{178}{3}\mathbf{c}_l \\ \frac{1468}{3}\mathbf{c}_l \\ \frac{16318}{3}\mathbf{c}_l \end{bmatrix}_{78 \times 1}$$

其中  $\mathbf{I}$  是  $13 \times 13$  的单位向量.

$$\mathbf{Aeq} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_1 & & \mathbf{0} \\ & & \mathbf{A}_1 & \mathbf{0} \\ & & & \mathbf{A}_1 & \mathbf{0} \end{bmatrix}_{28 \times 65}$$

$$\mathbf{Aeq} \cdot \mathbf{x} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_1 & & \mathbf{0} \\ & & \mathbf{A}_1 & \mathbf{0} \\ & & & \mathbf{A}_1 & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} f_{1l} \\ f_{2l} \\ f_{3l} \\ f_{4l} \\ z_l \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 f_{1l} \\ \mathbf{A}_1 f_{2l} \\ \mathbf{A}_1 f_{3l} \\ \mathbf{A}_1 f_{4l} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{bmatrix}_{28 \times 1} = \mathbf{beq}$$

$lb$  为  $65 \times 1$  的零向量,  $ub$  为空.



## 5 MATLAB 程序及运行结果

```

1 %MAU
2 f=[zeros(52,1);1];
3 A=[1,1,1,0,-1,0,0,0,0,0,0,0,0;
4     -1,0,0,1,0,0,0,0,-1,0,0,0,0;
5     0,-1,0,0,1,1,0,0,0,0,0,0,0;
6     0,0,-1,0,0,0,1,1,0,0,-1,0,0;
7     0,0,0,-1,0,0,-1,0,1,1,0,0,-1;
8     0,0,0,0,0,-1,0,-1,0,0,1,1,0;
9     0,0,0,0,0,0,0,0,0,-1,0,-1,1];
10 Aeq=[blkdiag(A,A,A,A),zeros(28,1)];
11 b1=[4,-4,0,0,0,0,0]';
12 b2=[4,0,-4,0,0,0,0]';
13 b3=[0,-4,4,0,0,0,0]';
14 b4=[4,0,0,0,0,0,-4]';
15 beq=[b1;b2;b3;b4];
16 I=eye(13);
17 c=5*ones(13,1);
18 a=[I,I,I,I,-c];
19 b=zeros(13,1);
20 lb=zeros(53,1);
21 x=linprog(f,a,b,Aeq,beq,lb,[]);
22 F=[x(1:13,:)]';
23     x(14:26,:)]';
24     x(27:39,:)]';
25     x(40:52,:)]'

```

程序运行结果如下：

$$\mathbf{f} = (f_{ml})_{4 \times 13} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 4 & 4 \\ 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \end{bmatrix}$$

$$\phi(\mathbf{f}) = 0.8$$

```

1  %M/M/1
2  f=[zeros(52,1);ones(13,1)];
3  A=[1,1,1,0,-1,0,0,0,0,0,0,0,0;
4     -1,0,0,1,0,0,0,0,-1,0,0,0,0;
5     0,-1,0,0,1,1,0,0,0,0,0,0,0;
6     0,0,-1,0,0,0,1,1,0,0,-1,0,0;
7     0,0,0,-1,0,0,-1,0,1,1,0,0,-1;
8     0,0,0,0,0,-1,0,-1,0,0,1,1,0;
9     0,0,0,0,0,0,0,0,0,-1,0,-1,1];
10 Aeq=[blkdiag(A,A,A,A),zeros(28,13)];
11 b1=[4,-4,0,0,0,0,0]';
12 b2=[4,0,-4,0,0,0,0]';
13 b3=[0,-4,4,0,0,0,0]';
14 b4=[4,0,0,0,0,0,-4]';
15 beq=[b1;b2;b3;b4];
16 I=eye(13);
17 T=[I;3*I;10*I;70*I;500*I;5000*I];
18 i=-[I;I;I;I;I;I];
19 a=[T,T,T,T,i];
20 c=5*ones(13,1);
21 b=[0*c;2/3*c;16/3*c;178/3*c;1468/3*c;16318/3*c];
22 lb=zeros(65,1);
23 x=linprog(f,a,b,Aeq,beq,lb,[]);
24 F=[x(1:13,:)]';
25     x(14:26,:)]';
26     x(27:39,:)]';
27     x(40:52,:)]';
28 sum(x(53:end,:))

```

程序运行结果如下：

$$f = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.1667 & 0 & 0.6667 & 3.3333 & 1.8333 & 0 & 3.5 & 0 & 1.6667 & 1.6667 & 1.6667 \\ 0 & 0 & 4 & 0 & 0 & 0 & 2.3333 & 1.6667 & 0 & 2.3333 & 0 & 1.6667 & 0 \end{bmatrix}$$

$$\phi(f) = 92.6667$$

## 6 结果分析

我们观察所得流分布矩阵，可以看出两种方案所得结果差异不是很大，将 MAU 方案和 M/M/1 方案转移的总流量分别加起来，MAU 方案共转移了 36 单位流量，而 M/M/1 方案转移了 35 单位流量，如果需要考虑转移流量的成本，M/M/1 方案要更为优异。

而我们可以看出，M/M/1 方案中的最大弧利用率也为 0.8，这和 MAU 方案的结果相同，而 MAU 方案中的 FT 成本函数值显然要比 M/M/1 方案中大，从这两个角度来说，M/M/1 方案都要比 MAU 方案好。

然而 M/M/1 方案中不等式约束更多，也需要花更多内存和时间来计算，在本例子中，MAU 方案的线性不等式约束矩阵  $\mathbf{A}$  大小为  $13 \times 53$ ，而 M/M/1 方案的  $\mathbf{A}$  矩阵大小为  $78 \times 65$ ，差异约为 6 倍，该差异来源于 FT 函数中多出的 6 个不等式，故在其他例子的测试中，该差异应该也会恒定在 6 倍左右。

综上，只考虑结果的优异性时，M/M/1 方案从各个方面都更好，但在大规模的测试问题中，我们就要考虑到计算成本的因素，可以选择更省时省力的 MAU 方案。