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 9 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 $K_1,K_2,\ldots,K_M$, 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：

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

b) 流平衡约束：

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

c) 流非负:

$$f_{ml}\ge 0, \quad m=1,\cdots,M,\forall l\in\mathcal{E} \tag{3}$$

令 $\boldsymbol{A}$ 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 $\boldsymbol{b}_m=(b_{m1},\cdots,b_{mN})^T,\boldsymbol{f}_m=(f_{m1},\ f_{m2},\cdots,\ f_{mE})^T$, 则可将等式约束 (2) 表示成

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

3 最优流量工程

若存在流分布 $\boldsymbol{f}$ 满足流量约束，我们应该思考怎样改变 $\boldsymbol{f}$ 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 $\phi(\boldsymbol{f})$ 来评价整体网络的堵塞情况，这里 $\phi(\boldsymbol{f})$ 是 $\boldsymbol{f}$ 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 $\phi(\boldsymbol{f})$ 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:

$$
\begin{aligned}
min \quad & \phi(\boldsymbol{f}, \boldsymbol{c}) && (4)\\
\text{s.t.} \quad & \boldsymbol{f} = \sum_{m=1}^{M} \boldsymbol{f}_m && (5)\\
& \boldsymbol{A}\boldsymbol{f}_m = \boldsymbol{b}_m, \quad m = 1, \cdots, M && (6)\\
& \boldsymbol{f}_m \geq \boldsymbol{0}, \qquad\quad m = 1, \cdots, M && (7)
\end{aligned}
$$

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

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

基于该目标函数的多商品流问题可转化为以下线性规划：

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

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 $\phi(\boldsymbol{f}) = \sum_{l\in\mathcal{E}} \phi(f_l)$, 其中

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

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:

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

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、目的节点信息如图如图**??**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 $\boldsymbol{c}^T = (5, 5, \cdots, 5)_{1\times 13}$

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

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

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：

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

4.1 MAU

在使用 MAU 方法时，需进行以下变形：

$$\boldsymbol{x}=\begin{bmatrix}\boldsymbol{f}_{1l}\\\boldsymbol{f}_{2l}\\\boldsymbol{f}_{3l}\\\boldsymbol{f}_{4l}\\z\end{bmatrix}_{53\times1},\qquad \boldsymbol{f}_{il}=\begin{bmatrix}f_{i1}\\f_{i2}\\f_{i3}\\\vdots\\f_{i13}\end{bmatrix}_{13\times1},\qquad \boldsymbol{f}=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}_{53\times1},\qquad \boldsymbol{f}^T\boldsymbol{x}=z$$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{I}, & \boldsymbol{I}, & \boldsymbol{I}, & \boldsymbol{I}, & -\boldsymbol{c}\end{bmatrix}_{13\times53},\qquad \boldsymbol{A}\cdot\boldsymbol{x}=\boldsymbol{f}_{1l}+\boldsymbol{f}_{2l}+\boldsymbol{f}_{3l}+\boldsymbol{f}_{4l}-z\cdot\boldsymbol{c}$$

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

$$Aeq=\begin{bmatrix}\boldsymbol{A}_1 & & & & \boldsymbol{0}\\ & \boldsymbol{A}_1 & & & \boldsymbol{0}\\ & & \boldsymbol{A}_1 & & \boldsymbol{0}\\ & & & \boldsymbol{A}_1 & \boldsymbol{0}\end{bmatrix}_{28\times53}$$

$$Aeq\cdot\boldsymbol{x}=\begin{bmatrix}\boldsymbol{A}_1 & & & & \boldsymbol{0}\\ & \boldsymbol{A}_1 & & & \boldsymbol{0}\\ & & \boldsymbol{A}_1 & & \boldsymbol{0}\\ & & & \boldsymbol{A}_1 & \boldsymbol{0}\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{f}_{1l}\\\boldsymbol{f}_{2l}\\\boldsymbol{f}_{3l}\\\boldsymbol{f}_{4l}\\z\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_1\boldsymbol{f}_{1l}\\\boldsymbol{A}_1\boldsymbol{f}_{2l}\\\boldsymbol{A}_1\boldsymbol{f}_{3l}\\\boldsymbol{A}_1\boldsymbol{f}_{4l}\end{bmatrix}=\begin{bmatrix}\boldsymbol{b}_1\\\boldsymbol{b}_2\\\boldsymbol{b}_3\\\boldsymbol{b}_4\end{bmatrix}_{28\times1}=beq$$

$\boldsymbol{lb}$ 为 53×1 的零向量，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:

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

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

其中 $\boldsymbol{I}$ 是 13×13 的单位向量.

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

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

$\boldsymbol{lb}$ 为 65×1 的零向量，ub 为空.

5 MATLAB 程序及运行结果

```
%MAU
f=[zeros(52,1);1];
A=[1,1,1,0,-1,0,0,0,0,0,0,0,0;
  -1,0,0,1,0,0,0,0,-1,0,0,0,0;
   0,-1,0,0,1,1,0,0,0,0,0,0,0;
   0,0,-1,0,0,0,1,1,0,0,-1,0,0;
   0,0,0,-1,0,0,-1,0,1,1,0,0,-1;
   0,0,0,0,0,-1,0,-1,0,0,1,1,0;
   0,0,0,0,0,0,0,0,0,-1,0,-1,1];
Aeq=[blkdiag(A,A,A,A),zeros(28,1)];
b1=[4,-4,0,0,0,0,0]';
b2=[4,0,-4,0,0,0,0]';
b3=[0,-4,4,0,0,0,0]';
b4=[4,0,0,0,0,0,-4]';
beq=[b1;b2;b3;b4];
I=eye(13);
c=5*ones(13,1);
a=[I,I,I,I,-c];
b=zeros(13,1);
lb=zeros(53,1);
x=linprog(f,a,b,Aeq,beq,lb,[]);
F=[x(1:13,:)';
   x(14:26,:)';
   x(27:39,:)';
   x(40:52,:)']
```

程序运行结果如下：

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

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

```
%M/M/1
f=[zeros(52,1);ones(13,1)];
A=[1,1,1,0,-1,0,0,0,0,0,0,0,0;
  -1,0,0,1,0,0,0,0,-1,0,0,0,0;
   0,-1,0,0,1,1,0,0,0,0,0,0,0;
   0,0,-1,0,0,0,1,1,0,0,-1,0,0;
   0,0,0,-1,0,0,-1,0,1,1,0,0,-1;
   0,0,0,0,0,-1,0,-1,0,0,1,1,0;
   0,0,0,0,0,0,0,0,0,-1,0,-1,1];
Aeq=[blkdiag(A,A,A,A),zeros(28,13)];
b1=[4,-4,0,0,0,0,0]';
b2=[4,0,-4,0,0,0,0]';
b3=[0,-4,4,0,0,0,0]';
b4=[4,0,0,0,0,0,-4]';
beq=[b1;b2;b3;b4];
I=eye(13);
T=[I;3*I;10*I;70*I;500*I;5000*I];
i=-[I;I;I;I;I;I];
a=[T,T,T,T,i];
c=5*ones(13,1);
b=[0*c;2/3*c;16/3*c;178/3*c;1468/3*c;16318/3*c];
lb=zeros(65,1);
x=linprog(f,a,b,Aeq,beq,lb,[]);
F=[x(1:13,:)';
   x(14:26,:)';
   x(27:39,:)';
   x(40:52,:)']
sum(x(53:end,:))
```

程序运行结果如下：

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

$$\phi(\boldsymbol{f}) = 92.6667$$

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 $\boldsymbol{A}$ 大小为 13×53，而 M/M/1 方案的 $\boldsymbol{A}$ 矩阵大小为 78×65，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。