现代管理科学方法（第3讲）

郭仁拥 博士/教授/博导

讲授内容

1. 交通分配问题的求解
2. 迭代步长的计算
3. 求解子问题的最短路算法
4. 课堂练习

1. 交通分配问题的求解

为了计算用户均衡流量分布方式，可求解如下最优化模型

$$\min_{\mathbf{x}\in\Omega_{\mathrm{L}}} V(\mathbf{x}) = \sum_{a\in L}\int_0^{x_a} c_a(x)\mathrm{d}x \qquad (1)$$

这里 $\Omega_{\mathrm{L}} \equiv \{\mathbf{x} \mid \mathbf{x}\in\Omega\}$，即 $\mathbf{x}$ 满足如下约束

$$x_a = \sum_{w\in W}\sum_{r\in R_w}\delta_{a,r} f_{r,w}, \quad \forall\ a\in L \qquad (2)$$

$$d_w = \sum_{r\in R_w} f_{r,w}, \quad \forall\ w\in W \qquad (3)$$

$$f_{r,w} \geq 0, \quad \forall\ r\in R_w,\ w\in W \qquad (4)$$

接下来介绍一类求解交通分配问题的Frank-Wolfe (FW)算法

该FW算法的计算过程如下：

第1步： 给定一个初始路段流量 $\mathbf{x}^{(0)} \in \Omega_{\mathrm{L}}$，设置 $k=0$.

第2步： 求解如下优化问题(5)来计算目标路段流量 $\mathbf{y}^{(k)}$

$$\min_{\mathbf{y}\in\Omega_{\mathrm{L}}} \mathbf{c}\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}} \mathbf{y} \tag{5}$$

即 $\mathbf{y}^{(k)} \in \arg\min_{\mathbf{y}\in\Omega_{\mathrm{L}}} \mathbf{c}\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}} \mathbf{y}$.

第3步： 确定迭代步长 $\alpha^{(k)}$（>0），使目标函数值 $V(\mathbf{x})$ 有某种意义的下降，令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\left(\mathbf{y}^{(k)} - \mathbf{x}^{(k)}\right)$.

第4步： 如果 $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| / \left\|\mathbf{x}^{(k)}\right\| < \varepsilon$，则停止迭代；否则，设置 $k=k+1$，返回第2步。

第4步中$\|\cdot\|$是一个欧式范数。例如，对于一个列向量$\mathbf{u}$，$\|\mathbf{u}\|=\sqrt{\mathbf{u}^{\mathrm{T}}\mathbf{u}}$。$\varepsilon$是一个较小的正数。

由于$\mathbf{x}^{(0)}\in\Omega_{\mathrm{L}}$，$\mathbf{y}^{(k)}\in\Omega_{\mathrm{L}}$，$\alpha^{(k)}\in(0,1]$，以及集合$\Omega_{\mathrm{L}}$是一个凸集，因此$\mathbf{x}^{(k+1)}=\left(1-\alpha^{(k)}\right)\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{y}^{(k)}\in\Omega_{\mathrm{L}}$，即流量迭代轨迹一直在集合$\Omega_{\mathrm{L}}$内。

第2步用来确定目标函数V的一个下降方向$\mathbf{y}^{(k)}-\mathbf{x}^{(k)}$。目标函数$V$的梯度方向可以刻画作

$$\nabla V(\mathbf{x})=(c_a(x_a),a\in L)^{\mathrm{T}}.$$

当流量迭代轨迹没有达到用户均衡时，由第2步中$\mathbf{y}^{(k)}$的定义可知

$$\nabla V\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}}\left(\mathbf{y}^{(k)}-\mathbf{x}^{(k)}\right)=\mathbf{c}\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}}\mathbf{y}^{(k)}-\mathbf{c}\left(\mathbf{x}^{(k)}\right)^{\mathrm{T}}\mathbf{x}^{(k)}<0.$$

因此$\mathbf{y}^{(k)}-\mathbf{x}^{(k)}$是目标函数$V$的一个下降方向

接下来说明$\left\|\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)}\right\| / \left\|\mathbf{x}^{(k)}\right\| < \varepsilon$可以作为一个算法迭代停止条件，使流量迭代轨迹达到用户均衡。当$\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}$时，$\mathbf{y}^{(k)}=\mathbf{x}^{(k)}$，即$\mathbf{x}^{(k)}$是优化问题(5)的最优解。因此，对应于$\mathbf{x}^{(k)}$的路径流量$\mathbf{f}^{(k)}$满足如下KKT条件

$$C_{r,w}\left(\mathbf{f}^{(k)}\right)=\mu_w+\lambda_{r,w}, \quad \forall\ r\in R_w,\ w\in W \tag{6}$$

$$\lambda_{r,w} f_{r,w}^{(k)}=0, \quad \lambda_{r,w}\geq 0, \quad f_{r,w}^{(k)}\geq 0, \quad \forall\ r\in R_w,\ w\in W \tag{7}$$

这里μ_w（$w\in W$）是对应守恒约束(3)的拉格朗日乘子，$\lambda_{r,w}$（$r\in R_w$, $w\in W$）是对应非负约束(4)的拉格朗日乘子。

结合公式(6)和(7)可得到

$$\left(C_{r,w}\left(\mathbf{f}^{(k)}\right)-\mu_w\right)f_{r,w}^{(k)}=0,\quad \forall\ r\in R_w,\ w\in W \tag{8}$$

$$C_{r,w}\left(\mathbf{f}^{(k)}\right)-\mu_w\geq 0,\quad f_{r,w}^{(k)}\geq 0,\quad \forall\ r\in R_w,\ w\in W \tag{9}$$

公式(8)和(9)意味着

$$C_{r,w}\left(\mathbf{f}^{(k)}\right)\begin{cases}=\mu_w, & \text{if } f_{r,w}^{(k)}>0,\forall r\in R_w, w\in W\\ \geq\mu_w, & \text{if } f_{r,w}^{(k)}=0,\forall r\in R_w, w\in W\end{cases}$$

也就是说 $\mathbf{f}^{(k)}$ 是一个UE流量方式。此外，优化问题(5)是一个凸优化问题（线性规划问题），因此 $\mathbf{x}^{(k)}$ 是优化问题(5)的最优解当且仅当 $\mathbf{f}^{(k)}$ 是一个UE流量方式。

2. 迭代步长的计算

设$V\left(\alpha^{(k)}\right)=V\left(\mathbf{x}^{(k)}+\alpha^{(k)}\left(\mathbf{y}^{(k)}-\mathbf{x}^{(k)}\right)\right)$，则有

$$\frac{dV\left(\alpha^{(k)}\right)}{d\alpha^{(k)}}=\nabla V\left(\mathbf{x}^{(k)}+\alpha^{(k)}\left(\mathbf{y}^{(k)}-\mathbf{x}^{(k)}\right)\right)^{\mathrm{T}}\left(\mathbf{y}^{(k)}-\mathbf{x}^{(k)}\right).$$

第3步中的迭代步长$\alpha^{(k)}$可由如下二分法计算：

第1步： 设置$\alpha^{(k)}=1/2$和$\beta=1/2$.

第2步： 设$\beta=\beta/2$.

第3步： 如果$dV\left(\alpha^{(k)}\right)/d\alpha^{(k)}>0$，则$\alpha^{(k)}=\alpha^{(k)}-\beta$；否则，$\alpha^{(k)}=\alpha^{(k)}+\beta$.

第4步： 如果$\beta<\varepsilon$，则算法迭代停止；否则，返回第2步。

3. 求解子问题的最短路算法

将约束 $\mathbf{y}=\mathbf{\Delta g}$ 代入优化问题(5)的目标函数，该优化问题可表示为仅包含决策变量 $\mathbf{g}$（路径流量）的形式：

$$\min_{\mathbf{g}} \mathbf{C}\left(\mathbf{f}^{(k)}\right)^{\mathrm{T}} \mathbf{g} \tag{10}$$

$$s.t. \quad \mathbf{d}=\mathbf{\Lambda g}, \quad \mathbf{g} \geq \mathbf{0} .$$

优化问题(10)进一步可以分解为一系列子问题

$$\min_{\mathbf{g}_w} \sum_{r \in R_w} C_{r,w}\left(\mathbf{f}^{(k)}\right) g_{r,w} \tag{11}$$

$$s.t. \quad d_w=\sum_{r \in R_w} g_{r,w}, \quad g_{r,w} \geq 0, \quad \forall\, r \in R_w,$$

对于任意 $w \in W$.

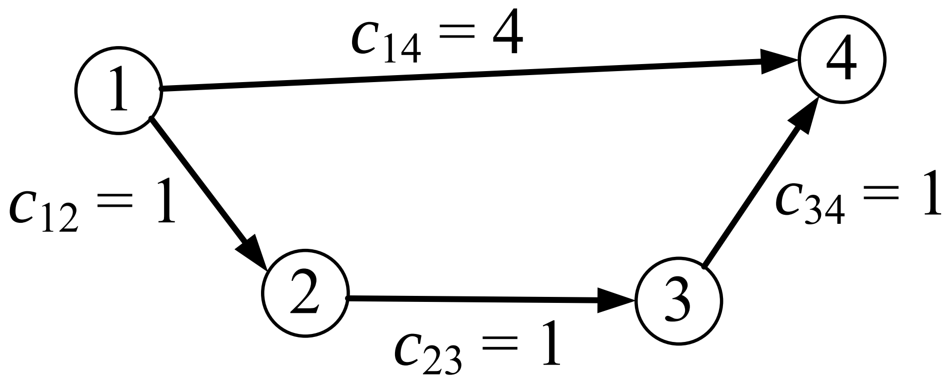
很显然，将OD对 w 间所有出行需求 d_w 分配到该OD对间具有最小出行费用的路径上（全有或全无分配，All-or-nothing assignment），就可以得到子问题(11)的最优解。因此，求解优化问题(5)可以转换为求解一些列最短路径问题。接下来介绍一类最短路算法—标号法（Label-correcting method），用来计算网络中一个起点到所有其他节点的最短路径。该算法以迭代方式扫描网络节点，每次迭代扫描发现更短的路径，当没有更短的路径被发现，算法结束。

- 每个路段由它的两个端点来表示，c_{ij} 是路段 ij 的长度（出行费用）
- 对于每个节点 i，两个信息被记录：**节点（当前）标号** l_i 和**（当前）前端节点** p_i
- **节点 i 的标号**记录根节点到节点 i 的（当前）最短路径距离
- **节点 i 的前端节点**记录（当前）最短路径上节点 i 的前端相邻节点
- **序列表**包含没有被检查和需要进一步检查的节点

- 初始时刻设置所有节点标号为正无穷，设置所有节点的前端节点为0，将起点 r 放入序列表并设置 $l_r = 0$
- 每次迭代开始从序列表选取一个节点 i 进行检查，测试所有与节点 i 通过一个路段相连的节点 j，如果 $l_i + c_{ij} < l_j$，则 $l_j = l_i + c_{ij}$， $p_j = i$，并且将节点 j 添加到序列表；经过所有相连的节点 j 被测试完，节点 i 的检查被完成，它被移出序列表

- 当序列表为空时，算法终止
- 依次追溯每个节点的前端节点，直到返回根节点，就可以找到根节点到其他任意节点的最短路径
- 从序列表移出的节点可能被再次添加到序列表

考虑右侧网络，该网络包含1个OD对和4个路段。使用标号法找出节点1到4的最短路径

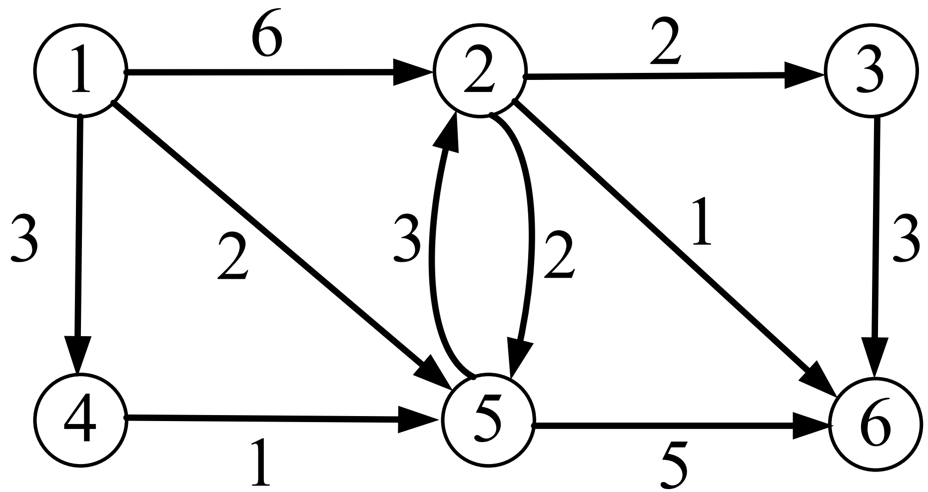


		标号表				前端节点表				
迭代	测试路段	节点1	节点2	节点3	节点4	节点1	节点2	节点3	节点4	序列表
0	——	0	∞	∞	∞	0	0	0	0	1
1	$1 \to 4$	0	∞	∞	4	0	0	0	1	1,4
2	$1 \to 2$	0	1	∞	4	0	1	0	1	2,4
3	$2 \to 3$	0	1	2	4	0	1	2	1	3,4
4	$3 \to 4$	0	1	2	3	0	1	2	3	4
5	——	0	1	2	3	0	1	2	3	

编程实现最短路算法的注意事项

1. 网络的存储：路段列表中路段由两端节点表示；为避免路段的重复搜索，具有相同前端节点的路段相邻存储；前端节点按升序安排

2. 序列表的管理：为避免重复计算，仅仅当一个节点不在序列表，才添加它进序列表；序列表被从上向下处理，一个从来不在序列表中的节点应该被添加到序列底部；一个之前在过序列表中的节点应该被放在序列顶端（即它是下一个待检查的节点）



前端节点	后端节点
1	2
1	4
1	5
2	3
2	6
2	5
3	6
4	5
5	2
5	6

在编程计算过程中，为了放一个节点在序列顶端，不需要物理移动所有节点位置；序列表由一个数组**s**来刻画，数组中每个元素描述了一个节点在序列表中的状态，即

$$s(i)=\begin{cases}-1 & \text{节点}i\text{之前在，但现在不在序列表中}\\ 0 & \text{节点}i\text{从来不在序列表中}\\ +j & \text{节点}i\text{在序列表中，}j\text{是相邻它的下一个节点}\\ +\infty & \text{节点}i\text{在序列表中，它是最后一个节点}\end{cases}$$

- 序列表顶端和低端由特定指针识别
- 当放节点 j 在序列顶端，设置 $s(j)=m$（这里 m 是之前序列中第一个节点），顶端指针指向 j
- 当放节点 j 在序列底部，设置 $s(j)=+\infty$ 和 $s(n)=j$（这里 n 是之前序列中最后一个节点），底部指针指向 j

最短路算法的计算过程如下：

第1步： 设置 $l_i = \infty$ 和 $p_i = 0$，$\forall\, i$；$l_o = 0$，下标 o 代表根节点；根节点 o 放入序列表

第2步： 如果序列表为空，则算法停止；否则，转到第3步

第3步： 将序列表中顶端节点记为 i，设 L_i 表示所有与节点 i 通过一个路段连接的后端节点集合

第4步： 如果集合 L_i 为空，则从序列表中移去节点 i，并转到第2步；否则，将集合 L_i 中下一个待测试节点记为 j

第5步： 如果 $l_i + c_{ij} < l_j$，则设置 $l_j = l_i + c_{ij}$ 和 $p_j = i$；否则，转到第4步

第6步： 如果节点 j 在序列集中，则转到第4步；否则，转到第7步

第7步： 如果节点 j 之前在过序列集，则将节点 j 放到序列集顶端，转到第4步；否则，将节点 j 放到序列集底部，转到第4步

在求解交通分配问题的FW算法中，初始路段流量 $\mathbf{x}^{(0)}$ 需要落在可行集 Ω_L 内，可以通过求解一个基于零流量路段费用 $\mathbf{c}(\mathbf{0})$ 的优化问题(5)来得到初始路段流量 $\mathbf{x}^{(0)}$，即求解如下优化问题

$$\min_{\mathbf{y} \in \Omega_L} \mathbf{c}(\mathbf{0})^T \mathbf{y}$$

换句话说，可以利用基于 $\mathbf{c}(\mathbf{0})$ 的全无全有分配得到初始路段流量 $\mathbf{x}^{(0)}$

为了计算用户均衡流量分布方式，需要求解如下优化问题

$$\min_{(\mathbf{x},\mathbf{f})} V(\mathbf{x}) = \sum_{a\in L}\int_0^{x_a} c_a(x)\mathrm{d}x$$

$$s.t.,\quad x_a = \sum_{w\in W}\sum_{r\in R_w}\delta_{a,r} f_{r,w},\quad \forall\ a\in L$$

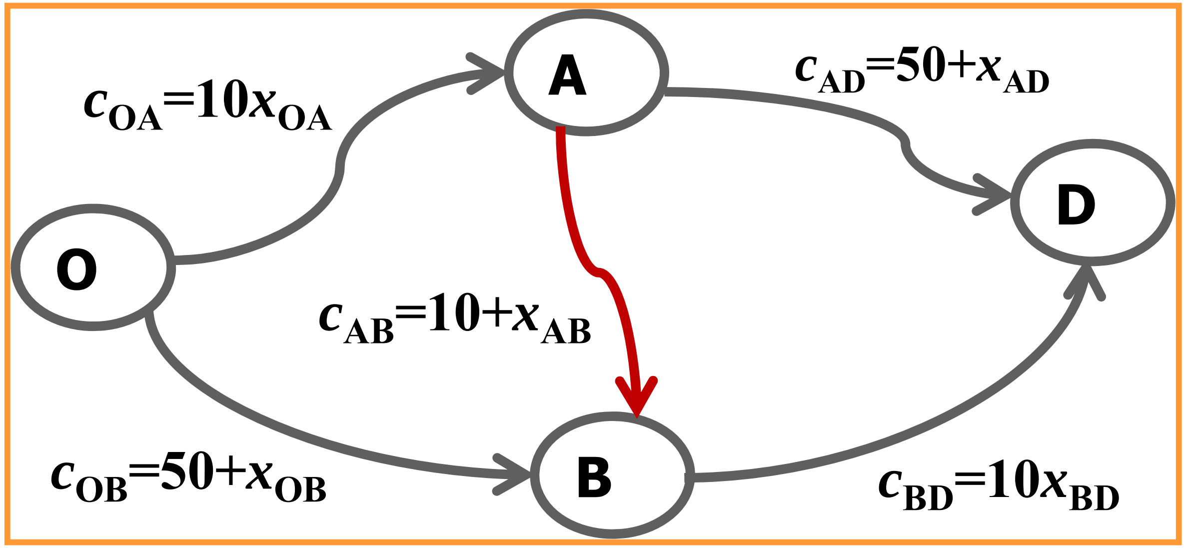
$$d_w = \sum_{r\in R_w} f_{r,w},\quad \forall\ w\in W$$

$$f_{r,w} \ge 0,\quad \forall\ r\in R_w,\ w\in W$$

以上优化问题的决策变量是路段流量 **x** 和路径流量 **f**

- 在一个现实的交通网络中，路径数量一般是远远大于路段数量
- FW算法仅仅需要存储和处理路段流量变量
- FW算法可应用于求解较大规模的现实交通分配问题

利用FW算法求解右侧网络的UE解，其中$d_{OD}=6$

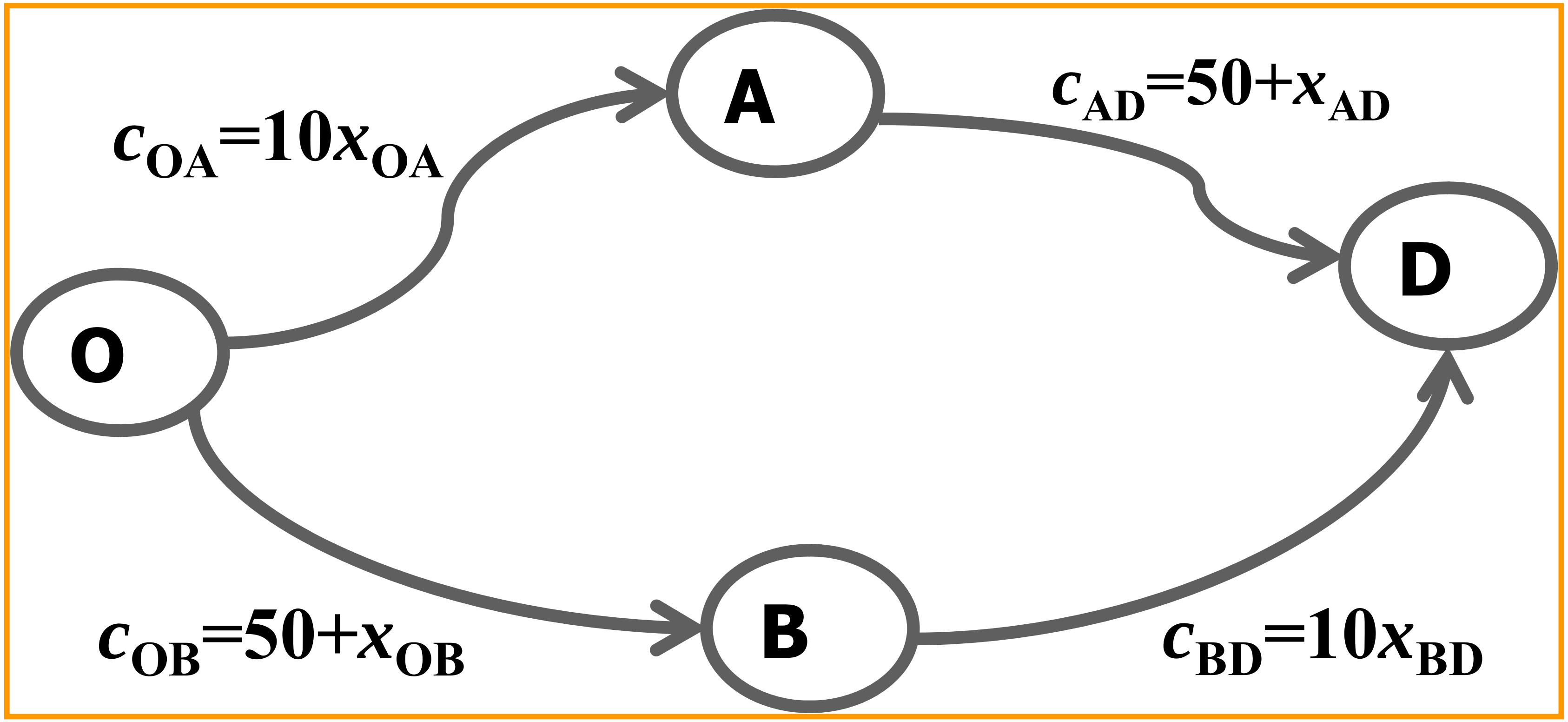


k	x_{OA}	x_{OB}	x_{AB}	x_{AD}	x_{BD}	y_{OA}	y_{OB}	y_{AB}	y_{AD}	y_{BD}	α	$V(\mathbf{x})$
0	6	0	6	0	6	6	0	0	6	0	0.361	438
1	6	0	3.83	2.17	3.83	0	6	0	0	6	0.309	409.83
2	4.14	1.86	2.65	1.50	4.50	6	0	0	6	0	0.088	387.72
3	4.31	1.69	2.41	1.90	4.10	0	6	0	0	6	0.054	386.67
4	4.08	1.92	2.28	1.79	4.21	6	0	0	6	0	0.036	386.31
5	4.15	1.85	2.20	1.94	4.06	0	6	0	0	6	0.025	386.15
6	4.04	1.96	2.15	1.90	4.10	6	0	0	6	0	0.018	386.08

k	x_{OA}	x_{OB}	x_{AB}	x_{AD}	x_{BD}	y_{OA}	y_{OB}	y_{AB}	y_{AD}	y_{BD}	α	$V(\mathbf{x})$
7	**4.08**	**1.92**	**2.11**	**1.97**	**4.03**	**0**	**6**	**0**	**0**	**6**	**0.013**	**386.04**
8	**4.02**	**1.98**	**2.08**	**1.94**	**4.06**	**6**	**0**	**0**	**6**	**0**	**0.010**	**386.02**
9	**4.04**	**1.96**	**2.06**	**1.98**	**4.02**	**0**	**6**	**0**	**0**	**6**	**0.007**	**386.01**
10	**4.01**	**1.99**	**2.05**	**1.97**	**4.03**	**6**	**0**	**0**	**6**	**0**	**0.005**	**386.01**
11	**4.02**	**1.98**	**2.04**	**1.99**	**4.01**	**0**	**6**	**0**	**0**	**6**	**0.004**	**386.00**
12	**4.01**	**1.99**	**2.03**	**1.98**	**4.02**	**6**	**0**	**0**	**6**	**0**	**0.003**	**386.00**
13	**4.01**	**1.99**	**2.02**	**1.99**	**4.01**	**0**	**6**	**0**	**0**	**6**	**0.002**	**386.00**
14	**4.00**	**2.00**	**2.02**	**1.99**	**4.01**	**6**	**0**	**0**	**6**	**0**	**0.002**	**386.00**
15	**4.01**	**1.99**	**2.01**	**2.00**	**4.00**	**0**	**6**	**0**	**0**	**6**	**0.001**	**386.00**
16	**4.00**	**2.00**	**2.01**	**1.99**	**4.01**	**6**	**0**	**0**	**6**	**0**	**0.001**	**386.00**
17	**4.00**	**2.00**	**2.01**	**2.00**	**4.00**	**0**	**6**	**0**	**0**	**6**	**0.001**	**386.00**
18	**4.00**	**2.00**	**2.01**	**2.00**	**4.00**	**6**	**0**	**0**	**6**	**0**	**0.001**	**386.00**
19	**4.00**	**2.00**	**2.00**	**2.00**	**4.00**	**0**	**6**	**0**	**0**	**6**	**0.000**	**386.00**
20	**4.00**	**2.00**	**2.00**	**2.00**	**4.00**	**6**	**0**	**0**	**6**	**0**	**0.000**	**386.00**

4. 课堂练习

请写出求解如下网络中用户均衡流量分布方式的优化模型，仿照上面算例写出FW算法的每次迭代计算结果（其中$d_{OD}=6$）



求解该网络中用户均衡流量分布方式的优化模型

$$\min_{(\mathbf{x},\mathbf{f})} V(\mathbf{x}) = \int_0^{x_{\mathrm{OA}}} c_{\mathrm{OA}}(x)dx + \int_0^{x_{\mathrm{AD}}} c_{\mathrm{AD}}(x)dx + \int_0^{x_{\mathrm{OB}}} c_{\mathrm{OB}}(x)dx + \int_0^{x_{\mathrm{BD}}} c_{\mathrm{BD}}(x)dx$$

$$s.t. \quad \begin{pmatrix} x_{\mathrm{OA}} \\ x_{\mathrm{AD}} \\ x_{\mathrm{OB}} \\ x_{\mathrm{BD}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_{\mathrm{OAD}} \\ f_{\mathrm{OBD}} \end{pmatrix}, \quad d_{\mathrm{OD}} = f_{\mathrm{OAD}} + f_{\mathrm{OBD}}, \quad f_{\mathrm{OAD}} \ge 0, \quad f_{\mathrm{OBD}} \ge 0$$

k	x_{OA}	x_{AD}	x_{OB}	x_{BD}	y_{OA}	y_{AD}	y_{OB}	y_{BD}	α	$V(\mathbf{x})$
0	**6**	**6**	**0**	**0**	**0**	**0**	**6**	**6**	**0.5**	**498**
1	**3**	**3**	**3**	**3**						**399**