Network Flow as a Partial Variable Problem

João Mota

October 26, 2014

The network flow problem is

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_{ij}) \\ \text{subject to} & x \geq 0 \\ & Bx = d\,, \end{array} \tag{1}$$

where B is the node-arc incidence matrix and d is the external inflow/outflow vector. This problem can be written as a distributed optimization problem with a partial variable. Consider the network in Fig. 1. The function at node 6 would be

$$f_6(x_{16}, x_{67}, x_{46}) = \phi_{16}(x_{16}) + \phi_{46}(x_{46}) + \phi_{67}(x_{67}) + \mathrm{i}_{\{x_{16}+x_{46}-x_{67}=d_6\}}(x_{16}, x_{67}, x_{46})\,.$$

The problem this node has to solve at each iteration is

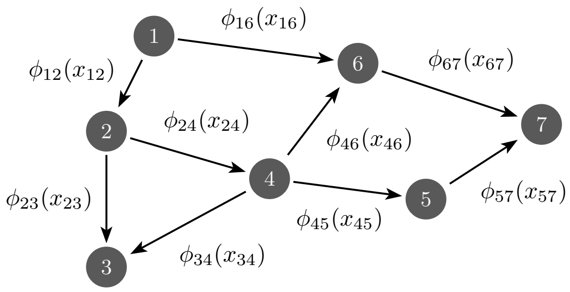

Figure 1: Example of a network flow problem. Each edge has a variable and a function of that variable associated. The goal is to minimize the sum of all functions while satisfying the flow constraints.

$$\begin{array}{ll} \underset{(x_{16},x_{46},x_{67})}{\text{minimize}} & \phi_{16}(x_{16}) + \phi_{46}(x_{46}) + \phi_{67}(x_{67}) + \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{46} \\ x_{67} \end{bmatrix} + \begin{bmatrix} x_{16} & x_{46} & x_{67} \end{bmatrix} \begin{bmatrix} a_1 & & \\ & a_2 & \\ & & a_3 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{46} \\ x_{67} \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{46} \\ x_{67} \end{bmatrix} = d_6 \\ & x_{16}, x_{46}, x_{67} \geq 0\,. \end{array} \tag{2}$$

Consider the multicommodity flow problem [1, Ch.17]:

$$\begin{array}{ll} \underset{x=\{x^k\}_{k=1}^K}{\text{minimize}} & \sum_{k=1}^K \sum_{(i,j)\in\mathcal{E}} \phi_{ij}^k(x_{ij}^k) \\ \text{subject to} & Bx^k = d^k\,, \quad k=1,\ldots,K \\ & 0 \leq \sum_{k=1}^K x_{ij}^k \leq c_{ij}\,, \quad k=1,\ldots,K,\ (i,j)\in\mathcal{E}\,, \end{array} \tag{3}$$

where x_{ij}^k is the flow of commodity k on edge (i,j) and $x^k = \{x_{ij}^k\}_{(i,j)\in\mathcal{E}}$, the variable of (3), is the collection of flows of commodity k along all the network edges. Also, d^k is the vector of input for commodity k. We can address this problem by considering the flows aggregated across commodities, i.e., the network does not distinguish between different commodities. To do so, define $\phi_{ij} = \sum_{k=1}^K \phi_{ij}^k$, $x_{ij} = \sum_{k=1}^K x_{ij}^k$, and $d = \sum_{k=1}^K d^k$. Now, simplify (3) to

$$\begin{array}{ll} \underset{x=\{x_{ij}\}_{(i,j)\in\mathcal{E}}}{\text{minimize}} & \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_{ij}) \\ \text{subject to} & Bx = d \\ & 0 \le x_{ij} \le c_{ij}\,, \quad (i,j)\in\mathcal{E}\,, \end{array} \tag{4}$$

which is the same as (1), plus the additional constraints $x_{ij} \le c_{ij}$. We model this problem as a congestion control problem by using

$$\phi_{ij}(x_{ij}) = \frac{x_{ij}}{c_{ij} - x_{ij}}\,,$$

where c_{ij} is the capacity of the (directed) edge $(i,j)\in\mathcal{E}$, as a model for the delay at edge (i,j) as a function of the aggregate rate of commodities at that edge, x_{ij}. This function is convex for $0 \le x \le c$:

$$\begin{aligned} \dot{\phi}_{ij}(x_{ij}) &= \frac{c_{ij} - x_{ij} + x_{ij}}{(c_{ij} - x_{ij})^2} = \frac{c_{ij}}{(c_{ij} - x_{ij})^2} \\ \ddot{\phi}_{ij}(x_{ij}) &= 2\frac{c_{ij}}{(c_{ij} - x_{ij})^3}\,, \end{aligned}$$

which is positive for $0 \le x_{ij} \le c_{ij}$. With this function, problem (2), for a generic node p, becomes

$$\begin{array}{ll} \underset{x=(x_1,\ldots,x_{n_p})}{\text{minimize}} & \sum_{i=1}^{n_p}\left(\frac{x_i}{c_i - x_i} + v_i x_i + a_i x_i^2\right) \\ \text{subject to} & b_p^\top x = d_p \\ & 0 \le x \le c\,. \end{array} \tag{5}$$

Since the projection onto the set $S := \{b_p^\top x = d_p\,,\, 0 \le x \le c\}$ is relatively simple, we will use the Barzilai-Borwein method, which requires the computation of the function and gradient of the objective (5) at each point and also a function that projects an arbitrary point y onto S.

Projection onto S. Let $y \in \mathbb{R}^q$ be given. The projection of y onto S is

$$\begin{array}{rll} \mathrm{P}(y) := & \underset{x}{\arg\min} & \frac{1}{2}\|x - y\|^2 \\ & \text{s.t.} & b^\top x = d \\ & & x \ge 0 \\ & & x \le c\,. \end{array} \tag{6}$$

Associating the Lagrange multipliers λ, μ, and η to the constraints of (6), respectively, the KKT equations are

$$\left\{\begin{array}{l} x - y + \lambda b - \mu + \eta = 0 \\ 0 \le x \le c \\ b^\top x = d \\ \mu \ge 0 \\ \eta \ge 0 \\ x^\top \mu = 0 \\ (x-c)^\top \eta = 0 \end{array}\right. \quad \Longleftrightarrow \quad \left\{\begin{array}{l} y - \lambda b = x + \eta - \mu \\ 0 \le x \le c \\ \mu \ge 0 \\ \eta \ge 0 \\ x^\top \mu = 0 \\ (x-c)^\top \eta = 0 \\ b^\top x = d \end{array}\right. . \tag{7}$$

We will now see that all the equations, but the last, imply that $x = \mathrm{P}_{[0,c]}(y - \lambda b)$ and $\eta - \mu = \mathrm{P}_{\mathbb{R}\backslash[0,c]}(y - \lambda b)$, where $\mathrm{P}_{[0,c]}(z)$ is the projection of z onto $[0,c]$, i.e., the ith component of z is given by:

$$\begin{cases} c_i & , z_i \geq c_i \\ 0 & , z_i \leq 0 \\ z_i & , \text{otherwise} \end{cases} .$$

To see that, we have to check that

$$(y - \lambda b - \mathrm{P}_{[0,c]}(y - \lambda b))^\top (s - \mathrm{P}_{[0,c]}(y - \lambda b)) \leq 0 \quad \Longleftrightarrow \quad (y - \lambda b - x)^\top (s - x) \leq 0\,,$$

for any $s \in [0, c]$. In fact, the first equation of (7) tells us that $y - \lambda b - x = \eta - \mu$. Thus,

$$\begin{aligned}
(\eta - \mu)^\top (s - x) &= (\eta - \mu)^\top s - \underbrace{\eta^\top x}_{=c^\top \eta} + \underbrace{\mu^\top x}_{=0} \\
&= (\eta - \mu)^\top s - c^\top \eta \\
&= \eta^\top s - \underbrace{\mu^\top s}_{\geq 0} - c^\top \eta \\
&\leq \underbrace{\eta^\top}_{\geq 0} \underbrace{(s - c)}_{\leq 0} \\
&\leq 0\,.
\end{aligned}$$

This shows that $x = \mathrm{P}_{[0,c]}(y - \lambda b)$ in (7). Therefore, that system of equations can be written as

$$\begin{cases} x = \mathrm{P}_{[0,c]}(y - \lambda b) \\ b^\top x = d \\ \eta - \mu = \mathrm{P}_{\mathbb{R} \setminus [0,c]}(y - \lambda b) \\ \mu \geq 0 \\ \eta \geq 0 \\ x^\top \mu = 0 \\ (x - c)^\top \eta = 0 \end{cases} .$$

Since we are only interested in finding x, we just need the first two equations.

We will now focus on finding λ. For that, replace x into $b^\top x = d$:

$$g(\lambda) := b_1 \mathrm{P}_{[0,c_1]}(y_1 - \lambda b_1) + b_2 \mathrm{P}_{[0,c_2]}(y_2 - \lambda b_2) + \cdots + b_q \mathrm{P}_{[0,c_q]}(y_q - \lambda b_q) = d\,.$$

First, note that each b_i is either $+1$ or -1.

- If $b_i = 1$, we have

$$b_i \mathrm{P}_{[0,c_i]}(y_i - \lambda b_i) = \begin{cases} c_i & , \lambda \leq y_i - c_i \\ y_i - \lambda & , y_i - c_i \leq \lambda \leq y_i \\ 0 & , \lambda \geq y_i \end{cases} .$$

- If $b_i = -1$, we have

$$b_i \mathrm{P}_{[0,c_i]}(y_i - \lambda b_i) = \begin{cases} 0 & , \lambda \leq -y_i \\ -y_i - \lambda & , -y_i \leq \lambda \leq c_i - y_i \\ -c_i & , \lambda \geq c_i - y_i \end{cases} .$$

Note that the range of g is bounded. To find λ such that $g(\lambda) = d$, note that g is a decreasing, piecewise-linear function. The points where it changes slope are, for all $i = 1, \ldots, q$,

- y_i and $y_i - c_i$ if $b_i = 1$;
- $-y_i$ and $c_i - y_i$ if $b_i = -1$.

Let z denote the above points after sorting,

$$z_1 \le z_2 \le \cdots \le z_{2q} \,,$$

and compute g for all the above points. Since g is decreasing,

$$g(z_1) \ge g(z_2) \ge \cdots \ge g(z_{2q}) \,.$$

If $d > g(z_1)$ or $g(z_{2q}) > d$, the problem (6) is not feasible. When feasible, we can find l such that $g(z_l) \ge d \ge g(z_{l+1})$. Since g is piecewise linear, we can find λ such that $g(\lambda) = d$ by interpolation:

$$\lambda = z_l + \frac{(z_{l+1} - z_l)(d - g(z_l))}{g(z_{l+1}) - g(z_l)} \,.$$

After finding λ as above, the solution to (6) can be easily computed as

$$\mathrm{P}(y) = \mathrm{P}_{[0,c]}(y - \lambda b) \,.$$

References

[1] R. Ahuja, T. Magnanti, and J. Orlin, *Network flows: Theory, algorithms, and applications*, Prentice Hall, 1993.