Network Optimization

Justin Chiu
Cornell Tech
jtc257@cornell.edu

September 29, 2021

Abstract

None

1 Network Traffic Problem

We consider the problem of maximizing network traffic. A network is a graph with V vertices v and E edges $e = (i, j)$. Each edge e has an associated capacity limit c_e. We get a set of requests, with each request r represented as source and target pairs (s, t),[1] plus a traffic demand d_r. The there are R requests. We would like to fulfill the demand for each request as much as possible, by splitting traffic across P_r valid paths. Each path associated with a request is a squence of edges given by $p = ((s, a), (a, b), \ldots, (z, t))$ from s to t. Our first constraint, constraint [1], ensures that each request has the constraint that the total traffic across associated paths $\sum_{p\in r} x_p \leq d_r$ does not exceed the request's demand. We say that $p \in r$ if p is a valid path from $s \to t$. We denote the total number of paths P. Additionally, we must ensure that edge traffic constraints hold. Multiple paths, as well as multiple requests, may result in overlapping traffic across particular edges, resulting in constraint [2]: $\sum_{p\in\pi(e)} x_p \leq c_e$, where $\pi(e) = \{p \mid e \in p\}$.

This yields the following optimization problem:

$$\begin{aligned}
\text{minimize} \quad & -\sum_r \sum_{p\in r} x_p \\
\text{subject to} \quad & \sum_{p\in r} x_p \leq d_r, \forall r \qquad [1] \\
& \sum_{p\in\pi(e)} x_p \leq c_e, \forall e \quad [2] \\
& x_p \geq 0, \forall p.
\end{aligned} \tag{1}$$

We derive the dual of this problem in order to more naturally incorporate constraints on the demands. Introducing dual variables λ_1 for constraint [1], λ_2 for constraint [2], and ν for the

[1] We assume there is only one unique request for each source and target pair.

nonnegativity of x, we have the Lagrangian:

$$\mathcal{L}(x,\lambda_1,\lambda_2,\nu) = -\sum_r \sum_{p\in r} x_p + \sum_r \lambda_{1r}\left(\sum_{p\in r} x_p - d_r\right) + \sum_e \lambda_{2e}\left(\sum_{p\in\pi(e)} x_p - c_e\right) + \nu^\top x. \quad (2)$$

The Lagrange dual function is given by

$$\begin{aligned} g(\lambda_1,\lambda_2,\nu) &= \inf_x \mathcal{L}(x,\lambda_1,\lambda_2,\nu) \\ &= -\sum_r \lambda_{1r} d_r - \sum_e \lambda_{2e} c_e + \inf_x \left(-\sum_r \sum_{p\in r} x_p + \sum_r \sum_{p\in r} \lambda_{1r} x_p + \sum_e \sum_{p\in\pi(e)} \lambda_{2e} x_p + \nu^\top x\right). \end{aligned}$$

The dual function is only greater than negative infinity if the last term is equal to 0. The dual of this problem is given by

$$\begin{array}{ll} \text{maximize} & -\sum_r \lambda_{1r} d_r - \sum_e \lambda_{2e} c_e \\ \text{subject to} & \sum_r 1(p\in r)\lambda_{1r} + \sum_e 1(p\in\pi(e))\lambda_{2e} + \nu_p = 1, \forall p \\ & \lambda_{1r} \geq 0, \forall r \\ & \lambda_{2e} \geq 0, \forall e, \end{array} \quad (3)$$

where we optimize over λ_1, λ_2.

We now propose a variant of the above problem with uncertain demands. In order to handle uncertainty, we introduce a risk tolerance, i.e. that the demand must fall within a particular range with probability $> \epsilon$, then optimize the worst-case demand constrained to that ellipse. To simplify the problem, we will assume that the demands are now random variables $d_r \sim \bar{d}_r + u_r$, where u_r has zero mean, meaning each demand is independent from the others. We denote the lower and upper ends of the uncertainty intervals as d_{rl}, d_{ru}.

Augmenting the dual problem above yields the optimization problem

$$\begin{array}{ll} \text{maximize} & \inf_d -\sum_r \lambda_{1r} d_r - \sum_e \lambda_{2e} c_e \\ \text{subject to} & \sum_r 1(p\in r)\lambda_{1r} + \sum_e 1(p\in\pi(e))\lambda_{2e} = 1, \forall p \\ & d_{rl} \leq d_r \leq d_{ru}, \forall r \\ & \lambda_{1r} \geq 0, \forall r \\ & \lambda_{2e} \geq 0, \forall e, \end{array} \quad (4)$$

where we optimize over λ_1, λ_2. Intuitively, if optimizing the dual problem gives us the best lower bound, then this gives us the worst best lower bound under the demand constraints.

2 Older version

While this form is compact, we would like the problem to separate over paths. This is not possible in Eqn. 1 because constraint [2] couples the traffic in paths across requests. In order to decouple this constraint, we will transform the problem by adding dummy variables. Afterwards, we will introduce slack variables to turn each of the resulting inequality constraints into equality constraints, which makes it more straightforward to apply the ADMM algorithm.

2.1 Dummy variables for edge traffic

We add one dummy variable z_{pe} for each combination of path and edge.[2] We then replace the x_p variables constraint [2] with dummy variables z_{pe}, resulting in constraint [3]. We also for $x_p = z_{pe}$ as constraint [4]. This results in the following problem:

$$\begin{array}{lll} \text{minimize} & -\sum_r \sum_{p \in r} x_p & \\ \text{subject to} & \sum_{p \in r} x_p \leq d_r, \forall r & [1] \\ & \sum_{p \in \pi(e)} z_{pe} \leq c_e, \forall e & [3] \\ & x_p - z_{pe} = 0, \forall e, p & [4] \\ & x_p \geq 0, \forall p & \\ & z_{pe} \geq 0, \forall e, p. & \end{array} \tag{5}$$

2.2 Slack variables for equality constraints

Finally, we add slack variables s_{1r}, s_{3e} to turn the inequality constraints into equality constraints.

$$\begin{array}{lll} \text{minimize} & -\sum_r \sum_{p \in r} x_p & \\ \text{subject to} & d_r - \sum_{p \in r} x_p - s_{1r} = 0, \forall r & [1] \\ & c_e - \sum_{p \in \pi(e)} z_{pe} - s_{3e} = 0, \forall e & [3] \\ & x_p - z_{pe} = 0, \forall e, p & [4] \\ & x_p \geq 0, z_{pe} \geq 0, s_{1r} \geq 0, s_{2e} \geq 0. & \end{array} \tag{6}$$

In order to write the Lagrangian for this problem, we introduce Lagrange multipliers $\lambda = (\lambda_1, \lambda_3, \lambda_4)$ for constraints [1], [3], and [4] respectively, where $\lambda_1 \in \mathbb{R}^{|R|}, \lambda_3 \in \mathbb{R}^{|E|}, \lambda_4 \in \mathbb{R}^{|P||E|}$.

[2] There are P_r paths for each request r, resulting in $E \sum_r \sum_{P_r}$ dummy variables.

The Lagrangian is then given by

$$\mathcal{L}_\rho(x, z, s, \lambda) = -\sum_r \sum_{p \in r} x_p + \lambda^\top F(x, z, s), \tag{7}$$

where $F(x, z, s) = (F_1, F_3, F_4)^\top$ and

$$[F_1]_r = d_r - \sum_{p \in r} x_p - s_{1r}, \quad [F_3]_e = c_e - \sum_{p \in \pi(e)} z_{pe} - s_{3e}, \quad [F_4]_{pe} = x_p - z_{pe}.$$

The augmented Lagrangian for this problem is

$$\mathcal{L}_\rho(x, z, s, \lambda) = -\sum_r \sum_{p \in r} x_p + \lambda^\top F(x, z, s) + (\rho/2)\|F(x, z, s)\|_2^2, \tag{8}$$

where we have added a quadratic term based on F and the hyperparameter $\rho \in \mathbb{R}_+$. This quadratic term is key to ADMM, as it must be 0 at the solution, resulting in a feasible solution for the original problem.

3 ADMM Updates

Given the augmented Lagrangian, the ADMM updates are as follows:

$$\begin{aligned}
x^{k+1} &:= \operatorname*{argmin}_x \mathcal{L}_\rho(x, z^k, s^k, \lambda^k) \\
z^{k+1} &:= \operatorname*{argmin}_z \mathcal{L}_\rho(x^{k+1}, z, s^k, \lambda^k) \\
s^{k+1} &:= \operatorname*{argmin}_s \mathcal{L}_\rho(x^{k+1}, z^{k+1}, s, \lambda^k) \\
\lambda^{k+1} &:= \lambda^k + \rho(F(x^{k+1}, z^{k+1}, s^{k+1})).
\end{aligned} \tag{9}$$

We solve for each update below.

3.1 Solving for x

We can compute $\operatorname{argmin}_x \mathcal{L}_\rho(x, z^k, s^k, \lambda^k)$ by restricting our attention to terms of $\mathcal{L}_\rho$ involving each x_p (for each path $p \in P$) and setting the derivative equal to 0:

$$\begin{aligned}
0 &= \nabla_{x_p}(-x_p + \lambda_{1r}(d_r - \sum_{p \in r} x_p - s_{1r}) + \sum_{e \in p} \lambda_{4pe}(x_p - z_{pe}) \\
&\quad + (\rho/2)((d_r - \sum_{p \in r} x_p - s_{1r})^2 + \sum_{e \in p}(x_p - z_{pe})^2)) \\
&= -1 - \lambda_{1r} + \sum_{e \in p} \lambda_{4pe} + \rho(-d_r + \sum_{p \in r} x_p + s_{1r}) + \rho \sum_{e \in p}(x_p - z_{pe}).
\end{aligned}$$

This gives us a system of equations for each request: $0 = A_r x_r + b_r$, where $x_r = (x_p)_{p\in r}$. Let P_r be the number of paths in request r. We can then solve for $x_r = -A_r^{-1} b_r$, where

$$\begin{aligned}
A_r &= \rho(\mathbf{1}_{P_r \times P_r} + \operatorname{diag}((|p|)_{p\in r})) \\
[b_r]_p &= -1 - \lambda_{1r} + \sum_{e\in p} \lambda_{4pe} + \rho(-d_r + s_{1r}) - \rho \sum_{e\in p} z_{pe}.
\end{aligned}$$

41 3.2 Solving for z

We perform a similar computation for $\operatorname{argmin}_z \mathcal{L}_\rho(x^{k+1}, z, s^k, \lambda^k)$:

$$\begin{aligned}
0 &= \nabla_{z_{pe}} \lambda_{3e}(c_e - \sum_{p'\in\pi(e)} z_{p'e} - s_{3e}) + \lambda_{4pe}(x_p - z_{pe}) \\
&\quad + (\rho/2)((c_e - \sum_{p'\in\pi(e)} z_{p'e} - s_e)^2 + (x_p - z_{pe})^2) \\
&= -\lambda_{3e} - \lambda_{4pe} + \rho(-c_e + \sum_{p'\in\pi(e)} z_{p'e} + s_e + z_{pe} - x_p).
\end{aligned}$$

This gives us a system of equations for each edge: $0 = A_e z_e + b_e$ allowing us to solve for $z_e = -A_e^{-1} b_e$. Let $P_e = |\pi(e)|$ be the number of paths that pass through edge e. We then have

$$\begin{aligned}
A_e &= \mathbf{1}_{P_e \times P_e} + I_{P_e \times P_e} \\
[b_e]_p &= -\lambda_{3e} - \lambda_{4pe} + \rho(-c_e + s_e - x_p).
\end{aligned}$$

42 3.3 Solving for s

Then, for $\operatorname{argmin}_{s_1} \mathcal{L}_\rho(x^{k+1}, z^{k+1}, s, y^k)$:

$$\begin{aligned}
0 &= \nabla_{s_{1r}} \lambda_{1r}(d_r - \sum_{p'\in r} x_{p'} - s_{1p}) + (\rho/2)((d_r - \sum_{p'\in r} x_{p'} - s_{1p})^2) \\
&= -\lambda_{1r} + \rho(-d_r + \sum_{p'\in r} x_{p'} + s_r) \\
s_{1r} &= \frac{\lambda_{1r} + \rho(d_r - \sum_{p'\in r} x_{p'})}{\rho},
\end{aligned}$$

and for $\operatorname{argmin}_{s_3} \mathcal{L}_\rho(x^{k+1}, z^{k+1}, s, y^k)$:

$$\begin{aligned}
0 &= \nabla_{s_{3e}} \lambda_{3e}(c_e - \sum_{p\in\pi(e)} z_{pe} - s_{3e}) + (\rho/2)((c_e - \sum_{p\in\pi(e)} z_{pe} - s_{3e})^2) \\
&= -\lambda_{3e} + \rho(-c_e + \sum_{p\in\pi(e)} z_{pe} + s_{3e}) \\
s_{3e} &= \frac{\lambda_{3e} + \rho(c_e - \sum_{p\in\pi(e)} z_{pe})}{\rho}.
\end{aligned}$$