· Rate control setup

n flows in a network, each passes over a fixed route

each flow has a nonnegative flow rate $f_1 \cdots f_n$

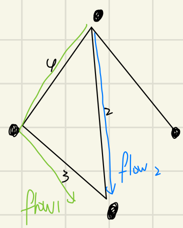
each flow has a utility function $U_j : \mathbb{R} \mapsto \mathbb{R}$, strictly concave and increasing

traffic t_i on link i is sum of flows passing through it

$t = Rf$ where R is the routing matrix

$$R_{ij} = \begin{cases} 1 & \text{if flow } j \text{ passes through link } i \\ 0 & \text{other wise} \end{cases}$$

link capacity constraint $t \le c$



$$R = \begin{array}{c|cc} & \text{flow1} & \text{flow2} \\ \hline 1 & & \\ 2 & & 1 \\ 3 & 1 & \\ 4 & 1 & \end{array}$$

· Rate control problem

$$\max.\ U(f) = \sum_j U_j(f_j)$$
$$\text{s.t.}\ \ Rf \le c$$

Convex problem: dual decomposition gives decentralized method

$$L(f, \lambda) = -U(f) + \lambda^T (Rf - c)$$
$$= -\lambda^T c + (R^T \lambda)^T f - U(f)$$
$$= -\lambda^T c + \sum_j r_j^T \lambda f_j - U_j(f_j)$$

$[\ r_j\ R\ ]\ [f]$

$[\ \lambda\ ][\ \ ][\ ]$

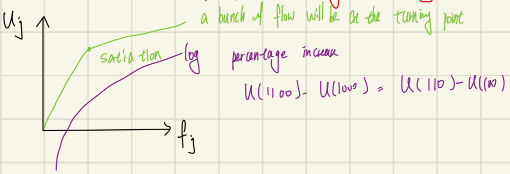
λ_i is the price (per unit flow) for using link i

$r_j^T \lambda$ is the sum of price along route j

$U_j(f_j) - r_j^T \lambda f_j$ = utility − cost = net utility

publish prices on all edges.

ask each flow to maximize the net utility without considering the link capacity



$$g(\lambda) = -\lambda^T c + \inf_f \sum_j r_j^T \lambda f_j - U_j(f_j)$$
$$= -\lambda^T c + \sup_f \sum_j -r_j^T \lambda f_j - -U_j(f_j)$$
$$= -\lambda^T c + \sum_j (-U_j)^* (-r_j^T \lambda)$$

the conjugate function

$$f^*(y) = \sup_x \left(y^T x - f(x) \right)$$

dual problem:
$$\max_\lambda.\ -\lambda^T c - \sum_j (-U_j)^* (-r_j^T \lambda)$$
$$\text{s.t.}\ \ \lambda \ge 0$$

how should the network price the links so that the charging is maximized when users take optimal policy

max. $g(\lambda) = -\lambda^T c + \inf_f \sum_j r_j^T \lambda f_j - U_j(f_j) = -\lambda^T c + \inf_f \lambda^T R f - U(f)$
s.t. $\lambda \geq 0$

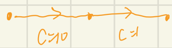
a subgradient for $-g(\lambda)$ is $\underbrace{c - R\bar{f}}_{\text{the trafic slackness}}$ $\left(\bar{f}_j = \operatorname{argmin}_{f_j} r_j^T \lambda f_j - U_j(f_j) \text{ s.t. } f_j \geq 0 \right)$

· Dual decomposition rate control algorithm

given initial link price vector $\lambda > 0$ (eg. $\lambda = 1$)
repeat

1. sum link price along each route $\Lambda_j = r_j^T \lambda$
2. optimize flows (separately) using flow prices $f_j := \operatorname{argmax} (U_j(f_j) - \Lambda_j f_j)$
3. calculate the link capacity margins $S := c - Rf$
4. $\lambda := (\lambda - \alpha_k S)_+$ if the margin is positive (there is slackness), then decrease the price for that link

(eg. when a link capacity constraint is slack, the corresponding lagrangian multiplier (price) is zero)



iterates can be (and often are) infeasible i.e. $Rf \nleq c$, but we do have feasibility in the limit
$g(\lambda)$ is a lower bound on minus utility $U(f)$

· Generating feasible flows

define $\eta_i = (Rf)_i / c_i$ ($\eta_i < 1$ means link i is under capacity)

define f^{feas} as
$$f_j^{feas} = \frac{f_j}{\max \{ \eta_i \mid \text{flow } j \text{ passes over link } i \}}$$

· Single Commodity network flow setup

Connected, directed graph with n links, p nodes

variable x_j denotes flow (traffic) on arc j

given external souce (or sink) flow s_i at node i $\quad (\mathbf{1}^T s = 0)$

node incidence matrix $A \in \mathbb{R}^{p \times n}$ is

$$A_{ij} = \begin{cases} 1 & \text{arc } j \text{ enters } i \\ -1 & \text{arc } j \text{ leaves node } i \\ 0 & \text{otherwise} \end{cases} \qquad A = \begin{array}{c} \text{arc} \\ \begin{bmatrix} & 1 \\ +1 & -1 \\ -1 & \end{bmatrix} \text{node} \end{array}$$

flow conservation $\quad Ax + s = 0 \quad$ (kcl) $\quad$ the nullspace of A $(Ax = 0)$ is circulation

$$\phi(x) = \sum_j^n \phi_j(x_j)$$ is separable convex flow cost function

· Network flow Problem

optimal single commodity network flow problem

$$\min. \ \sum_{j=1}^n \phi_j(x_j)$$

$$\text{s.t.} \quad Ax + s = 0$$

(dual variable will be potential at the nodes)

Network flow Lagrangian

$$\mathcal{L}(x,v) = \phi(x) + v^T(\overbrace{Ax - s}^{\text{kcl violation}})$$

$$[\ v^T\ ]\begin{bmatrix} A \\ \cdot \end{bmatrix}\begin{bmatrix} x \end{bmatrix} \qquad A = \begin{array}{c} \text{arcs} \\ \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix} \text{nodes} \end{array}$$

$$= v^T s + \phi(x) + (A^T v)^T x$$

$$= v^T s + \sum_j \left[\phi_j(x_j) + (a_j^T v) x_j\right]$$

separable in x

$a_j^T v$ is the difference of potential across arc j

if we add a constant to every element of v.

the lagrangian does not change $\begin{cases} v^T s \text{ doesn't change because } \mathbf{1}^T s = 0 \\ a_j^T v \text{ doesn't change} \end{cases}$

Network flow dual

$$g(v) = \inf_x \mathcal{L}(x,v)$$

$$= v^T s + \inf_x \left[\sum_j \phi_j(x_j) + (a_j^T v) x_j\right] \qquad f^*(y) = \sup_x \left(y^T x - f(x)\right)$$

$$= v^T s - \sum_j \phi_j^*(-a_j^T v)$$

$$= v^T s - \sum_j \phi_j^*(-\Delta v_j)$$

dual problem $\quad \max_v \ v^T s + \sum_j \phi_j(\tilde{x}_j) + (a_j^T v)\tilde{x}_j \quad$ where $\quad \tilde{x}_j = \operatorname{argmin}_x \ \phi_j(x) + (a_j^T v) x$

· Recovering primal from dual

Strictly convex ϕ_j means unique minimizer $x_j^*(y)$ of $\phi_j(x_j) - y x_j$

if ϕ_j is differential $x_j^*(y) = (\phi_j')^{-1}(y)$ inverse of derivative function

optimal flows, from optimal potentials : $x_j^* = x_j^*(-\Delta V_j^*)$ the nonlinear IV characteristic

a subgradient of the negative dual

$$-g(v) = -v^T s - \phi(\bar{x}) - v^T A \bar{x} \qquad \text{where } \bar{x} = \arg\min_x \phi(x) + v^T A x$$

$$-A\bar{x} - s \in \partial(-g)(v)$$

· Dual decomposition network flow algorithm

given initial potential vector v

repeat

1. determine link flows from potential difference

   $x := \arg\min \phi(x) + v^T A x$
2. compute flow surplus at each node

   $g = Ax + s$
3. update node potentials

   $v := v + \alpha_k g$

if the receiving node have too much current going in, then the potential of the node should go up