# Chapter 7

# Flow Models in Routing and Congestion Control

(Revised, November 06, -B. Hajek)

This chapter gives an introduction to the study of routing and congestion control in networks, using flow models and convex optimization. The role of routing is to decide which routes through the network should be used to get the data from sources to destinations. The role of congestion control is to determine data rates for the users so that fairness among users is achieved, and a reasonable operating point is found on the tradeoff between throughput and delay or loss. The role of flow models is to give a simplified description of network control protocols, which focus on basic issues of stability, rates of convergence, and fairness. Elements of routing and congestion control not covered in this chapter, include routing tables, messages exchanged for maintaining routing tables, packet headers, and so on.

## 7.1 Convex functions and optimization

Some basic definitions and facts concerning convex optimization are briefly described in this section, for use in the rest of the chapter.

Suppose that $\Omega$ is a subset of $\mathbb{R}^n$ for some $n \geq 1$. Also, suppose that $\Omega$ is a *convex set*, which by definition means that for any pair of points in $\Omega$, the line segment connecting them is also in $\Omega$. That is, whenever $x, x' \in \Omega$, $ax + (1-a)x' \in \Omega$ for $0 \leq a \leq 1$. A function $f$ on $\Omega$ is a *convex function* if along any line segment in $\Omega$, $f$ is less than or equal to the value of the linear function agreeing with $f$ at the endpoints. That is, by definition, $f$ is a convex function on $\Omega$ if $f(ax+(1-a)x') \leq af(x)+(1-a)f(x')$ whenever $x, x' \in \Omega$ and $0 \leq a \leq 1$. A *concave function* $f$ is a function $f$ such that $-f$ is convex, and results for minimizing convex functions can be translated to results for maximizing concave functions.

The *gradient*, $\nabla f$, of a function $f$ is a vector valued function, defined to be the vector of first partial derivatives of $f$:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

We assume that $\nabla f$ exists and is continuous. If $x \in \Omega$ and if $v$ is a vector such that $x + \epsilon v \in \Omega$ for small enough $\epsilon > 0$, then the *directional derivative* of $f$ at $x$ in the direction $v$ is given by

$$\lim_{\epsilon \to 0} \frac{f(x+\epsilon v) - f(x)}{\epsilon} = (\nabla f(x)) \cdot v = \sum_i \frac{\partial f}{\partial x_i}(x) v_i \tag{7.1}$$

We write $x^* \in \arg\min_{x \in \Omega} f(x)$ if $x^*$ minimizes $f$ over $\Omega$, i.e. if $x^* \in \Omega$ and $f(x^*) \leq f(x)$ for all $x \in \Omega$.

**Proposition 7.1.1** *Suppose $f$ is a convex function on a convex set $\Omega$ with a continuous gradient function $\nabla f$. Then $x^* \in \arg\min_{x\in\Omega} f(x)$ if and only if $(\nabla f(x^*)) \cdot (x - x^*) \geq 0$ for all $x \in \Omega$.*

**Proof.** (Necessity part.) Suppose $x^* \in \arg\min_{x\in\Omega} f(x)$ and let $x \in \Omega$. Then $x^* + \epsilon(x - x^*) \in \Omega$ for $0 \leq \epsilon \leq 1$, so by the assumption on $x^*$, $f(x^*) \leq f(x^* + \epsilon(x - x^*))$, or equivalently, $\frac{f(x^*+\epsilon(x-x^*))-f(x^*)}{\epsilon} \geq 0$. Letting $\epsilon \to 0$ and using (7.1) with $v = x - x^*$ yields $(\nabla f(x^*)) \cdot (x - x^*) \geq 0$, as required.

(Sufficiency part.) Suppose $x^* \in \Omega$ and that $(\nabla f(x^*)) \cdot (x - x^*) \geq 0$ for all $x \in \Omega$. Let $x \in \Omega$. Then by the convexity of $f$, $f(\epsilon x + (1-\epsilon)x^*)) \leq \epsilon f(x) + (1-\epsilon) f(x^*)$ for $0 \leq \epsilon \leq 1$. Equivalently, $f(x) \geq f(x^*) + \frac{f(x^*+\epsilon(x-x^*))-f(x^*)}{\epsilon}$. Taking the limit as $\epsilon \to 0$ and using the hypotheses about $x^*$ yields that $f(x) \geq f(x^*) + (\nabla f(x^*)) \cdot (x - x^*) \geq f(x^*)$. Thus, $x^* \in \arg\min_{x\in\Omega} f(x)$. ■

## 7.2 The Routing Problem

The problem of routing in the presence of congestion is considered in this section. The flow rates of the users are taken as a given, and the problem is to determine which routes should provide the flow. In a later section, we consider the joint routing and congestion control problem, in which both the flow rates of the users and the routing are determined jointly. The following notation is used for both problems.

$J$ is the set links.

$R$ is the set of routes. Each route $r$ is associated with a subset of the set of links. We write $j \in r$ to denote that link $j$ is in route $r$. If all routes use different sets of links, we could simply consider routes to be subsets of links. But more generally we allow two different routes to use the same set of links.

$A$ is the link-route incidence matrix, defined by $A_{j,r} = 1$ if $j \in r$ and $A_{j,r} = 0$ otherwise.

$S$ is the set of users. Each user $s$ is associated with a subset of the set of routes, which is the set of routes that serve user $s$. We write $r \in s$ to denote that route $r$ serves user $s$. We require that the sets of routes for different users br disjoint, so that each route serves only one user.

$H$ is the user-route incidence matrix, defined by $H_{s,r} = 1$ if $r \in s$ and $H_{s,r} = 0$ otherwise.

$y_r$ is the flow on route $r$.

$f_j$ is the flow on a link $j$: $f_j = \sum_{r:j \in r} y_r$, or in vector form, $f = Ay$.

$x_s$ is the total flow available to user $s$: $x_s = \sum_{r \in s} y_r$, or in vector form, $x = Hy$. The vector $x = (x_s : s \in S)$ is fixed for the routing problem, and variable for the joint congestion control and routing problem considered in Section 7.4.

$D_j(f_j)$ is the cost of carrying flow $j$ on link $j$. The function $D_j$ is assumed to be a convex, continuously differentiable, and increasing function on $\mathbb{R}_+$. The cost associated with flow rate $f_j$ on link $j$ is $D_j(f_j)$.

The routing problem, which we now consider, is to specify the route rates for given values of the user rates. It can be written as:

*ROUTE(x,H,A,D):*

$$\min \sum_{j \in J} D_j(\sum_{r:j \in r} y_r)$$

subject to

$$x = Hy$$

over

$$y \geq 0.$$

We say that the vector $y$ is a feasible flow meeting demand $x$ if $y \geq 0$ and $x = Hy$. Problem *ROUTE(x,H,A,D)* thus concerns finding a feasible flow $y$, meeting the demand $x$, which minimizes the total cost $F(y)$, defined by

$$F(y) = \sum_j D_j(\sum_{r:j \in r} y_r).$$

As far as the mathematics is concerned, the links could be resources other than connections between nodes, and the set of links in a route could be an arbitrary subset of $J$. But to be concrete, we can think of there being an underlying graph with nodes representing switches in a network, links indexed by ordered pairs of nodes, and routes forming paths through the network. We could assume each user has a specific origin node and destination node, and the routes that serve a user each connect the source node to the destination node of the user.

A possible choice for the function $D_j$ would be

$$D_j(f_j) = \frac{f_j}{C_j - f_j} + f_j d_j \tag{7.2}$$

where $C_j$ is the capacity of link $j$ and $d_j$ is the propagation delay on link $j$. If $f_j$ and $C_j$ are in units of packets per second, if arrivals are modeled as a Poisson process, and if service times are modeled as exponentially distributed (based on random packet lengths with an exponential distribution), then a link can be viewed as an $M/M/1$ queue followed by a fixed delay element, and $\frac{1}{C_j - f_j}$ is the mean system time for the queue. Thus, the mean transit time of a packet over the link $j$, including

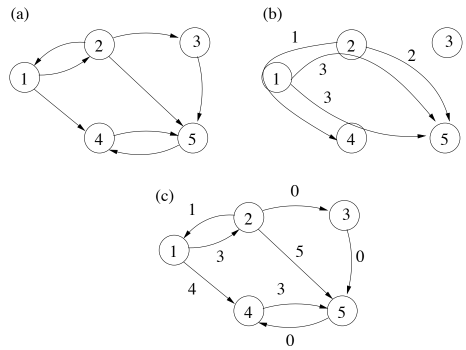



Figure 7.1: Example of a network showing (a) the network, (b) route flows, and (c) link flows.
.

the propagation delay, is $\frac{1}{C_j - f_j} + d_j$. Then by Little's law, $D_j(f_j)$ given by (7.2), represents the mean number of packets in transit on link $j$. Therefore, the total cost, $F(y)$, represents the mean number of packets in transit in the network. By Little's law, the average delay of packets entering the network is thus $F(y)/\sum_s x_s$. For fixed user flow rates $(x_s : s \in S)$, minimizing the cost is therefore equivalent to minimizing the delay in the network, averaged over all packets.

**Example** Consider the network graph shown in Figure 7.1(a). For this example there is an underlying set of 5 nodes, numbered 1 through 5. For this example we take

$J = \{(1,2),(1,4),(2,1),(2,3),(2,5),(3,5),(4,5),(5,4)\}$, with each link labeled by its endpoints.

$S = \{(1,5),(2,4),(2,5)\}$, with each user labeled by its origin and destination nodes.

We assume user $(1,5)$ is served by routes (1,2,5), (1,4,5), (1,2,3,5)
user $(2,4)$ is served by routes (2,1,4), (2,5,4)
user $(2,5)$ is served by routes (2,5), (2,3,5)
where each route is labled by the sequence of nodes that it traverses.

For brevity we write the flow $y_r$ for a route $r = (1,2,5)$ as $y_{125}$, rather than as $y_{(1,2,5)}$. We also write the rate $x_s$ for a user $s = (1,5)$ as $x_{1,5}$, rather than as $x_{(1,5)}$, and the flow $f_j$ for a link $j = (1,2)$ as $f_{1,2}$, rather than as $f_{(1,2)}$. An example of an assignment of flow rates to routes, shown in Figure 7.1(b), is given by

$$y_{125} = 3.0,\ y_{145} = 3.0,\ y_{1235} = 0, \quad y_{214} = 1.0,\ y_{254} = 0, \quad y_{25} = 2.0,\ y_{235} = 0 \tag{7.3}$$

which implies the flow rates to the users

$$x_{1,5} = 6.0, \quad x_{2,4} = 1.0, \quad x_{2,5} = 2.0 \tag{7.4}$$

and the link rates $f_{1,2} = 3$, $f_{1,4} = 4$, $f_{2,1} = 1$, etc., as shown in Figure 7.1(c). The resulting cost is

$$F(y) = D_{1,2}(3) + D_{1,4}(4) + D_{2,1}(1) + D_{2,3}(0) + D_{2,5}(5) + D_{3,5}(0) + D_{4,5}(3) + D_{5,4}(0).$$

We consider, in the context of this example, the effect on the cost of changing the flow rates. If $y_{125}$ were increased to $y_{125} + \epsilon$ for some $\epsilon > 0$, then the cost $F(y)$ would increase, with the amount of the increase given by

$$\begin{aligned} \triangle F &= D_{1,2}(3+\epsilon) - D_{1,2}(3) + D_{2,5}(5+\epsilon) - D_{2,5}(5) \\ &= \epsilon(D'_{1,3}(3) + D'_{2,5}(5)) + o(\epsilon) \end{aligned}$$

Similarly, if $y_{145}$ were increased by $\epsilon$, the cost would increase by $\epsilon(D'_{1,4}(4) + D'_{4,5}(3))\epsilon + o(\epsilon)$. If instead, $y_{145}$ were decreased by $\epsilon$, the cost would decrease by $\epsilon(D'_{1,4}(4) + D'_{4,5}(3))\epsilon + o(\epsilon)$. If two small changes were made, the effect on $F(y)$, to first order, would be the sum of the effects. Thus, if $y_{125}$ were increased by $\epsilon$ and $y_{145}$ were simultaneously decressed by $\epsilon$, that is, if $\epsilon$ units of flow were *deviated* to route $(1, 2, 5)$ from route $(1, 4, 5)$, then the change in cost would be

$$\text{(cost of 145}\rightarrow\text{125 flow deviation)} = \epsilon(D'_{1,2}(3) + D'_{2,5}(5) - D'_{1,4}(4) - D'_{4,5}(3)) + o(\epsilon)$$

That is, to first order, the change in cost due to the flow deviation is $\epsilon$ times the first derivative length $D'_{1,2}(3) + D'_{2,5}(5)$ of route $(1, 2, 5)$ minus the first derivative length $D'_{1,4}(4) + D'_{4,5}(3)$ of route $(1, 4, 5)$. Thus, for small enough $\epsilon$, the flow deviation will decrease the cost if flow is deviated from a longer route to a shorter one, if the lengths of the routes are given by the sums of the first derivative link lengths, $D_j(f_j)$, of the links in the routes.

We return from the example to the general routing problem. The observation made about flow deviation in the example is general. It implies that a necessary condition for $y$ to be optimal is that for any $s$ and any $r \in s$, the rate $y_r$ is strictly positive only if $r$ has minimum first derivative length among the routes serving $s$. Since the set of feasible rate vectors $y$ meeting the demand $x$ is a convex set, and the cost function $F$ is convex, it follows from elementary facts about convex optimization, described in Section 7.1, that the first order necessary condition for optimality is also sufficient for optimality. Specifically, the following is implied by Proposition 7.1.1.

**Proposition 7.2.1** *A feasible flow $y = (y_r : r \in R)$ meeting demand $x$ minimizes the cost over all such flows, if and only if there exists a vector $(\lambda_s : s \in S)$ so that $\sum_{j \in r} D'_j(f_j) \geq \lambda_s$, with equality if $y_r > 0$, whenever $r \in s \in S$.*

Given a flow vector $y$, let $\delta_r$ denote the first derivative length of route $r$. That is, $\delta_r = \sum_{j \in r} D_j(\sum_{r':j \in r'} y_{r'})$. Note that if the rate requirement for a user $s$ is strictly positive, i.e. $x_s > 0$, then the parameter $\lambda_s$ in Proposition 7.2.1 must satisfy $\lambda_s = \min_{r \in s} \delta_r$. The example and Proposition 7.2.1 suggest the following algorithm, called the flow-deviation algorithm, for finding an optimal flow $y$. The algorithm is iterative, and we describe one iteration of it.

**The flow deviation algorithm** One iteration is described. At the beginning of an iteration, a feasible flow vector $y$, meeting the specified demand $x$, is given. The iteration finds the next such vector, $\overline{y}$.

Step1. Compute the link flow $f_j$ and first derivative link lengths, $D_j(f_j)$, for $j \in J$.

Step 2. For each user $s$, find a route $r_s^*$ serving $s$ with the minimum first derivative link length.

Step 3. Let $\alpha \in [0,1]$. For each user $s$, deviate a fraction $\alpha$ of the flow from each of the other routes serving $s$ to $r_s^*$. That is, for each $s$, let

$$\overline{y}_r = \begin{cases} (1-\alpha)y_r & \text{if } r \in s \text{ and } r \neq r_s^* \\ y_r + \alpha(\sum_{r':r\neq r' \in s} y_{r'}) & \text{if } r = r_s^* \end{cases}$$

Adjust $\alpha$ to minimize $F(\overline{y})$. The resulting vector $\overline{y}$ is the result of the iteration.

The flow deviation algorithm is not a distributed algorithm, but some parts of it can be implemented in a distributed way. A user could gather the first derivative link lengths of its routes by having the links along each route signal the information in passing packets. A user could then determine which of its routes has the shortest first derivative link length. Synchronization of the flow deviation step across users, and in particular, the fact that all users must collectively determine a single best value of $\alpha$, is the main obstacle to distributed implementation of the flow deviation algorithm presented. However, by having each user respond slowly enough, possibly asynchronously, convergence can be achieved.

It may give insight into the problem to ignore the step size problem, find an algorithm, and then address the step size selection problem later. This suggests describing an algorithm by an ordinary differential equation. For the routing problem at hand, imagine that all users gradually shift their flow towards routes with smaller first derivative link length. Let $y(t)$ denote the vector of flows at time $t$, and let $\delta_r(t)$ denote the corresponding first derivative link length of route $r$ at time $t$, for each $r \in R$. Assume that the users adjust their rates slowly enough that the variables $\delta_r(t)$ can be continuously updated. In the following, we suppress the variable $t$ in the notation, but keep in mind that $y$ and $(\delta_r : r \in R)$ are time varying. We've seen that if $r$ and $r'$ are two routes for a given user $s$, if $\delta_{r'} > \delta_r$, and if $y_{r'} > 0$, then the cost $F(y)$ can be reduced by deviating flow from $r'$ to $r$. If the speed of deviation is equal to $y_{r'}(\delta_{r'} - \delta_r)$, then $y_{r'}$ is kept nonnegative, and the factor $(\delta_{r'} - \delta_r)$ avoids discontinuities in case two routes have nearly the same length. Suppose such changes are made continuously for all pairs of routes. Let $\theta = (\theta_{r,r'} : r, r' \in R)$ be a matrix of constants such that $\theta_{r,r'} = \theta_{r',r} > 0$ whenever $r, r' \in s$ for some $s \in S$, and $\theta_{r,r'} = 0$ if $r$ and $r'$ serve different users. We will use the constant $\theta_{r,r'}$ as the relative speed for deviation of flow between routes $r$ and $r'$. The following equation gives a flow deviation algorithm in ordinary differential equation form.

**A continuous flow deviation algorithm**

$$\dot{y}_r = \sum_{r' \in s} \left\{ \theta_{r',r} y_{r'} (\delta_{r'} - \delta_r)_+ - \theta_{r,r'} y_r (\delta_r - \delta_{r'})_+ \right\} \quad \text{for } r \in s \in S \tag{7.5}$$

Because the changes in $y$ are due to transfers between routes used by the same user, we expect $x(t) = Hy(t)$, to be constant. Indeed, using the symmetry of $\theta$ and interchanging the variables $r$

and $r'$ on a summation, yields that for any user $s$,

$$\begin{aligned}\dot{x}_s = \sum_{r\in s} \dot{y}_r &= \sum_{r\in s}\sum_{r'\in s} \theta_{r',r} y_{r'}(\delta_{r'}-\delta_r)_+ - \sum_{r\in s}\sum_{r'\in s} \theta_{r,r'} y_r(\delta_r-\delta_{r'})_+ \\ &= \sum_{r\in s}\sum_{r'\in s} \theta_{r',r} y_{r'}(\delta_{r'}-\delta_r)_+ - \sum_{r\in s}\sum_{r'\in s} \theta_{r',r} y_{r'}(\delta_{r'}-\delta_r)_+ \\ &= 0\end{aligned}$$

Now $\frac{\partial F}{\partial y_r} = \delta_r$, so by the chain rule of differentiation, and the same interchange of $r$ and $r'$ used in computing $\dot{x}_s$,

$$\begin{aligned}\frac{dF(y(t))}{dt} &= \sum_r \delta_r \dot{y}_r \\ &= \sum_{r\in R}\sum_{r'\in R} \left\{\theta_{r',r} y_{r'}\delta_r(\delta_{r'}-\delta_r)_+ - \theta_{r,r'} y_r \delta_r(\delta_r-\delta_{r'})_+\right\} \\ &= \sum_{r\in R}\sum_{r'\in R} \left\{\theta_{r',r} y_{r'}\delta_r(\delta_{r'}-\delta_r)_+ - \theta_{r',r} y_{r'} \delta_{r'}(\delta_{r'}-\delta_r)_+\right\} \\ &= -\sum_{r\in R}\sum_{r'\in R} \theta_{r',r} y_{r'}(\delta_{r'}-\delta_r)_+^2 \\ &\leq 0\end{aligned}$$

Thus, the cost $F(y(t))$ is decreasing in time. Moreover, the rate of decrease is strictly bounded away from zero away from a neighborhood of the set of $y$ satisfying the necessary conditions for optimality. It follows that all limit points of the trajectory $(y(t) : t \geq 0)$ are solutions to the routing problem, and $F(y(t))$ converges monotonically to the minimum cost.

The step size problem is an important one, and is not just a minor detail. If the users do not implement some type of control on how fast they make changes, oscillations could occur. For example, one link could be lightly loaded one minute, so the next, many users could transfer flow to the link, to the point it becomes overloaded. Then they may all transfer flow away from the link, and the link may be very lightly loaded again, and so on. Such oscillations have limited the use of automated dynamic routing based on congestion information. In applications, an important facet of the implementation is that the flow rates at the links often have to be estimated, and the accuracy of an estimate over an interval of length $t$ is typically on the order of $1/\sqrt{t}$, by the central limit theorem.

The two routing algorithms considered in this section are considered to be *primal* algorithms, because the flow rates to be found are directly updated. In contrast, in a dual algorithm, the primal variables are expressed in terms of certain dual variables, or prices, and the algorithm seeks to identify the optimal values of the dual variables. Dual algorithms arise naturally for networks with hard link constraints of the form $f_j \leq C_j$, rather than with the use of soft link constraints based on cost functions $D_j(f_j)$ as in this section. See the remark in Section 7.7

## 7.3 Utility Functions

Suppose a consumer would like to purchase or consume some commodity, such that an amount $x$ of the commodity has utility or value $U(x)$ to the consumer. The function $U$ is called the utility function. Utility functions are typically nondecreasing, continuously differentiable, concave functions on $\mathbb{R}_+$. Commonly used examples include:

(i) $U(x) = \beta \log x$ (for some constant $\beta > 0$)

(ii) $U(x) = \left(\frac{\beta}{1-\alpha}\right)(x^{1-\alpha} - 1)$ (for some constants $\alpha, \beta > 0, \alpha \neq 1$)

(iii) $U(x) = \frac{\beta}{2}\left[\bar{x}^2 - (\bar{x} - x)_+^2\right]$ (for some constants $\beta, \bar{x} > 0$)

(iv) $U(x) = -\beta \exp(-\gamma x)$ (for some constant $\gamma > 0$)

The function in (ii) converges to the function in (i) as $\alpha \to 1$.

If $x$ is increased by a small amount $\epsilon$ the utility $U(x)$ is increased by $\epsilon U'(x) + o(x)$. Thus $U'(x)$ is called the *marginal utility* per unit of commodity. Concavity of $U$ is equivalent to the natural property that the marginal utility is monotone nonincreasing in $x$.

If a rational consumer can buy commodity at a price $\lambda$ per unit, then the consumer would choose to purchase an amount $x$ to maximize $U(x) - x\lambda$. Differentiation with respect to $x$ yields that the optimal value $x_{opt}$ satisfies

$$U'(x_{opt}) \leq \lambda, \quad \text{with equality if } x_{opt} > 0. \tag{7.6}$$

It is useful to imagine increasing $x$, starting from zero, until the consumer's marginal utility is equal to the price $\lambda$. If $U$ is strictly concave, and if $(U')^{-1}$ is the inverse function of $U'$, we see that the response of the consumer to price $\lambda$ is to purchase $x_{opt} = (U')^{-1}(\lambda)$ units of commodity.

The response function for a logarithmic utility function $U(x) = \beta \log x$ has a particularly simple form, and an important interpretation. The response of a rational consumer with such a utility function to a price $\lambda$ is to purchase $x_{opt}$ given by $x_{opt} = \beta/\lambda$. So $\lambda x_{opt} = \beta$ for any $\lambda > 0$. That is, such a consumer effectively decides to make a payment of $\beta$, no matter what the price per unit flow $\lambda$ is.

*Exercise:* Derive the response functions for the other utility functions given in (i)-(iv) above.

## 7.4 Joint Congestion Control and Routing

In Section 7.2, the flow requirement $x_s$ for each user $s$ is assumed to be given and fixed. However, if the values of the $x_s$'s are too large, then the congestion in the network, measured by the sum of the link costs, may be unreasonably large. Perhaps no finite cost solution exists. Congestion control consists of restricting the input rates to reasonable levels. One approach to doing this fairly is to allow the $x_s$'s to be variables; and to assume utility functions $U_s$ are given. The joint congestion control and routing problem is to select the $y_r$'s. and hence also the $x_s$'s. to maximize the sum of the utilities of the flows minus the sum of link costs. We label the problem the system problem, because it involves elements of the system ranging from the users' utility functions to the topology and cost functions of the network. It's given as follows:

*SYSTEM(U,H,A,D) (for joint congestion control and routing with soft link constraints):*

$$\max \sum_{s\in S} U_s(x_s) - \sum_{j\in J} D_j(\sum_{r:j\in r} y_r)$$

subject to

$$x = Hy$$

over

$$x, y \geq 0.$$

This problem involves maximizing a concave objective function over a convex set. Optimality conditions, which are both necessary and sufficient for optimality, can be derived for this joint congestion control and routing problem in a fashion similar to that used for the routing problem alone.[1] The starting point is to note that the change in the objective function due in increasing $x_r$ by $\epsilon$, for some user $s$ and some route $r$ serving $s$, is

$$(U_s'(x_s) - \sum_{j\in r} D_j'(f_j))\epsilon + o(\epsilon)$$

This implies that if $y$ is an optimal flow, then

$$U_s'(x_s) \leq \sum_{j\in r} D_j'(f_j), \quad \text{with equality if } y_r > 0, \text{ whenever } r \in s \in S. \tag{7.7}$$

That is, condition (7.7) is necessary for optimality of $x$. Due to the concavity of the objective function, the optimality condition (7.7) is also sufficient for optimality, and indeed Proposition 7.1.1 implies that (7.7) is both necessary and sufficient for optimality.

The first derivative length of a link $j$, $D_j'(f_j)$, can be thought of as the marginal price (which we will simply call price, for brevity) for using link $j$, and the sum of these prices over $j \in r$ for a route $r$, is the price for flow along the route. In words, (7.7) says the following for the routes $r$ serving a user $s$: The price of any route with positive flow is equal to the marginal utility for $s$, and the price of any route with zero flow is greater than or equal to the marginal utility for $s$.

## 7.5 Hard Constraints and Prices

In some contexts the flow $f$ on a link is simply constrained by a capacity: $f \leq C$. This can be captured by a cost function which is zero if $f \leq C$ and infinite if $f > C$. However, such a cost function is highly discontinuous. Instead, we could consider for some small $\epsilon$, $\delta > 0$ the function

$$D(f) \quad = \quad \frac{1}{2\epsilon}(f - (C-\delta))_+^2.$$

The price $D'(f)$ is given by

$$D'(f) \quad = \quad \frac{1}{\epsilon}(f - (C-\delta))_+. \tag{7.8}$$

[1] In fact, through the introduction of an additional imaginary link, corresponding to undelivered flow, it is possible to reduce the joint congestion control and routing problem described here to a pure routing problem. See [1, Section 6.5.1].

As $\epsilon, \delta \to 0$, the graph of the price function $D'$ converges to a horizontal segment up to $C$ and a vertical segment at $C$, as shown in Figure 7.2. That is, for the hard constraint $f \leq C$, the

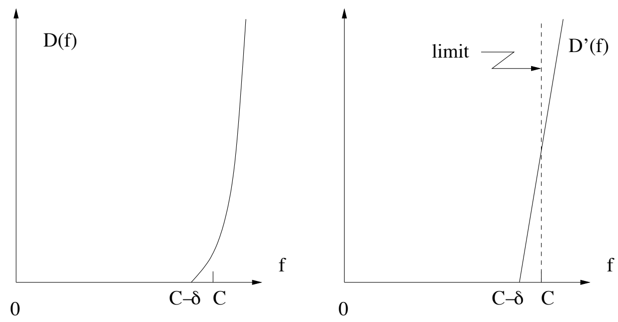



Figure 7.2: Steep cost function, approximating a hard constraint

price is zero if $f < C$ and the price is an arbitrary nonnegative number if $f = C$. (This is the complementary slackness condition for Kuhn-Tucker constraints.)

The routing problem formulated in Section 7.2 already involves hard constraints–not inequality constraints on the links–but equality constraints on the user rates $x_s$. The variable $\lambda_s$ can be viewed as a marginal price for the rate allocation to user $s$. The rate $x_s$ of user $s$ could be increased by $\epsilon$ at cost $\lambda_s \epsilon + o(\epsilon)$ by increasing the flow along one of the routes $r$ serving $s$ with minimum first derivative link length $\lambda_s$.

The joint routing and congestion control problem with hard link constraints is similar to problem *SYSTEM(U,H,A,D)*, but with the cost functions $(D_j : j \in J)$ replaced by a vector of capacities, $C = (C_j : j \in J)$:

*SYSTEM(U,H,A,C) (for joint congestion control and routing with hard link constraints):*

$$\max \sum_{s \in S} U_s(x_s)$$

subject to

$$x = Hy, \quad Ay \leq C$$

over

$$x, y \geq 0.$$

The optimality conditions for *SYSTEM(U,H,A,C)* are, for some vector $\mu = (\mu_j : j \in J)$,

$$\mu_j \geq 0, \quad \text{with equality if } f_j < C_j, \quad \text{for } j \in J \tag{7.9}$$

$$U_s'(x_s) \leq \sum_{j \in r} \mu_j, \quad \text{with equality if } y_r > 0, \quad \text{whenever } r \in s \in S. \tag{7.10}$$

Equation (7.10) is identical to (7.7), except $\mu_j$ replaces $D_j'(f_j)$. Equation (7.9) shows that $\mu_j$ satisfies the limiting behavior of the approximating function $D'(f_j)$ in (7.8), as $\epsilon, \delta \to 0$. It is plausible that (7.9) and (7.10) give optimality conditions for *SYSTEM(U,H,A,C)*, because in a formal sense they are the limit of (7.7) as the soft constraints approach hard constraints. The sufficiency of these conditions for optimality is proved in an exercise, based on use of a Lagrangian.

## 7.6 Decomposition into Network and User Problems

A difficulty of the joint congestion control and routing problem is that it involves the users' utility functions, the topology of the network, and the link cost functions or capacities. Such large, heterogeneous optimization problems can often be decoupled, using prices (a.k.a. Lagrange multipliers) across interfaces. A particularly elegant formulation of this method was given in the work of Kelly [3], followed here. See Figure 7.3. The idea is to view the problem faced by a user as how much to

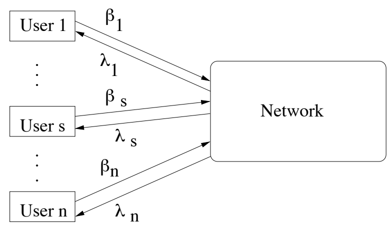



Figure 7.3: Coupling between the network and user problems in the system decomposition

pay. Once each user $s$ has decided to pay $\beta_s$ for its rate, the network mechanisms can ignore the real utility function $U_s(x_s)$ and pretend the utility function of user $s$ is the surrogate utility function $V_s(x_s) = \beta_s \log(x_s)$. That is because, as noted in Section 7.3, the response of a user with utility function $\beta_s \log(x_s)$ to a fixed price $\lambda_s$ is to pay $\beta_s$, independently of the price. The network can determine $(x, y)$, using the surrogate valuation functions. In addition, for each user $s$, the network can determine a price $\lambda_s$, equal to the minimum marginal cost of the routes serving $s$:

$$\lambda_s = \min_{r \in s} \delta_r \quad \text{where } \delta_r = \begin{cases} \sum_{j \in r} D_j(f_j) & \text{in the case of soft link constraints} \\ \sum_{j \in r} \mu_j & \text{in the case of hard link constraints.} \end{cases} \tag{7.11}$$

The network reports $\lambda_s$ to each user $s$. User $s$ can then also learn the allocation rate $x_s$ allocated to it by the network, because $x_s = \lambda_s \beta_s$. The other side of the story is the user optimization problem. A user $s$ takes the price $\lambda_s$ as fixed, and then considers the possibility of submitting a new value of the payment, $\beta_s$, in response. Specifically, if the price $\lambda_s$ resulting from payment $\beta_s$ is less than the marginal utility $U_s'(\beta_s/\lambda_s)$, then the user should spend more (increase $\beta_s$). Note that the network and user optimization problems are connected by the payments declared by the users (the $\beta_s$'s) and the prices declared by the network (the $\lambda_s$'s). The network need not know the utility functions of the users, and the users need not know the topology or congestion cost functions of the network.

Input data for the network problem includes the vector of payments $\beta$. but not the utility functions. The problem is as follows.

*NETWORK(β,H,A,D) (for joint congestion control and routing with soft link constraints):*

$$\max \sum_{s\in S} \beta_s \log(x_s) - \sum_{j\in J} D_j(\sum_{r:j\in r} y_r)$$

subject to

$$x = Hy$$

over

$$x, y \geq 0.$$

The user problem for a given user $s$ does not involve the network topology. It is as follows.

*USER$_s$($U_s, \lambda_s$):*

$$\max U_r\left(\frac{\beta_s}{\lambda_s}\right) - \beta_s$$

over

$$\beta_s \geq 0.$$

Let us prove that if the optimality conditions of both the user problem and the network problem are satisfied, then the optimality conditions of the system problem are satisfied. We first consider the problem with soft link constraints. The optimality conditions for the user problem are:

$$U_s'(x_s) \leq \lambda_s, \quad \text{with equality if } x_s > 0, \text{ for } s \in S. \tag{7.12}$$

The optimality conditions for the network problem are given by (7.7) with $U_s'$ replaced by $V_s'$. Since $V_s'(x_s) = \frac{\beta_s}{x_s}$ (for this to always be true, we interpret $\frac{\beta_s}{x_s}$ to be zero if $\beta_s = x_s = 0$), the optimality conditions for the network problem are thus:

$$\frac{\beta_s}{x_s} \leq \sum_{j\in r} D_j'(f_j), \quad \text{with equality if } y_r > 0, \text{ whenever } r \in s \in S. \tag{7.13}$$

The fact, true by (7.12), that $U_s'(x_s) \leq \lambda_s$ for all $s \in S$, and the rule (7.11) for selecting $\lambda_s$, imply that $U_s'(x_s) \leq \sum_{j\in r} D_j'(f_j)$ whenever $r \in s \in S$. Thus, it remains to prove the equality condition in (7.7) in case $y_r > 0$. If $r \in s \in S$ and $y_r > 0$, then $x_s > 0$. Therefore, by (7.13),

$$\lambda_s = \frac{\beta_s}{x_s} \tag{7.14}$$

Also, equality holds in both (7.12) and (7.13), so using (7.14) to eliminate $\lambda_s$, we find equality holds in (7.7).

The story is similar for the case of hard link constraints, as we see next. The network problem is

*NETWORK(β,H,A,C) (for joint congestion control and routing with hard link constraints):*

$$\max \sum_{s \in S} \beta_s \log(x_s)$$

subject to

$$x = Hy, \quad Ay \leq C$$

over

$$x, y \geq 0.$$

and the same user problem is used. The optimality condition for the user problem is again (7.12). The optimality condition for the network problem is given by (7.9) and (7.10) with $U'_s$ replaced by $V'_s$:

$$\mu_j \geq 0, \quad \text{with equality if } f_j < C_j, \quad \text{for } j \in J \tag{7.15}$$

$$\frac{\beta_s}{x_s} \leq \sum_{j \in r} \mu_j, \quad \text{with equality if } y_r > 0, \quad \text{whenever } r \in s \in S. \tag{7.16}$$

We wish to show that the user problem optimality conditions, (7.12), the network problem optimality conditions, (7.15) and (7.16), and the choice of $(\lambda_s : s \in S)$, given in (7.11), imply the optimality conditions for the system problem, (7.9) and (7.10). Equation (7.15) is the same as (7.9). The proof of (7.16) follows by the same reasoning used for soft constraints.

In summary, whether the link constraints are soft or hard, if the network computes the prices using $\lambda_s = \min_{r \in s} \delta_r$, and if the optimality conditions of the network and user problems are satisfied, then so are the optimality conditions of the system problem.

## 7.7 Specialization to pure congestion control

A model for pure congestion control is obtained by assuming the users in the joint congestion control and routing problem are served by only one route each. In this situation we can take the set of routes $R$ to also be the set of users., and write $x_r$ for the rate allocated to user $r$. If hard link constraints are used, the congestion control problem is the following.

*SYSTEM(U,A,D) (for congestion control with soft link constraints):*

$$\max \sum_{r \in R} U_r(x_r) - \sum_{j \in J} D_j(\sum_{r : j \in r} x_r)$$

over

$$x \geq 0.$$

The optimality conditions for this problem are

$$U'_r(x_r) \leq \sum_{j \in r} D'_j(f_j), \quad \text{with equality if } x_r > 0, \text{ for all } r \in R. \tag{7.17}$$

If hard link constraints are used, the congestion control problem is the following.

*SYSTEM(U,A,C) (for congestion control with hard link constraints):*

$$\max \sum_{r\in R} U_r(x_r)$$

subject to

$$Ax \le C$$

over

$$x \ge 0.$$

The optimality conditions for this problem are

$$\mu_j \ge 0, \quad \text{with equality if } f_j < C_j, \quad \text{for } j \in J \tag{7.18}$$

$$U'_r(x_r) \le \sum_{j\in r} \mu_j, \quad \text{with equality if } x_r > 0, \quad \text{for } r \in R. \tag{7.19}$$

The corresponding network problem is obtained by replacing the utility function $U_r(x_r)$ by $\beta_r \log(x_r)$ for each $r$:

*NETWORK(β,A,C) (for congestion control with hard link constraints):*

$$\max \sum_{r\in R} \beta_r \log(x_r)$$

subject to

$$Ax \le C$$

over

$$x \ge 0.$$

The optimality conditions for this problem are special cases of (7.18) and (7.19) and are given by (with $f = Ax$):

$$\mu_j \ge 0, \quad \text{with equality if } f_j < C_j, \text{ for } j \in J \tag{7.20}$$

$$\beta_r = x_r \cdot (\sum_{j\in r} \mu_j), \quad \text{for } r \in R, \tag{7.21}$$

The user problem for the congestion control system problem is the same as for the joint routing and congestion control problems. Basically, the user cannot distinguish between these two problems.

**Remark** We remark briefly about dual algorithms, which seek to find an optimal value for the vector of dual variables, $\mu$. The primal variables, $x$, are determined by $\mu$ through (7.21). The dual variable $\mu_j$ is adjusted in an effort to make sure that $f_j \le C_j$, and to satisfy the other optimality condition, (7.20). Intuitively, if for the current value $\mu(t)$ of the dual variables, the constraint $f_j \le C_j$ is violated, $\mu_j$ should be increased. If the contraint $f_j$ is strictly smaller than $C_j$, then $\mu_j$

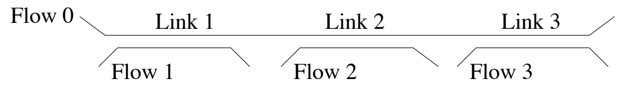



Figure 7.4: A three link network with four users

should be decreased (if it is strictly positive) or be held at zero otherwise.

**Example** Consider the following instance of the problem NETWORK($\beta$, A,C), for congestion control with hard link constraints. Suppose $J = \{1, 2, 3\}$ and $R = \{0, 1, 2, 3\}$, so there are three links and fours users, with one route each. Suppose the route of user 0 contains all three links, and the route of user $i$ contains only link $i$, for $1 \leq i \leq 3$, as shown in Figure 7.4. Thus, the optimality conditions are given by (7.20) and (7.21). In order for $x$ to be an optimal flow, it is necessary for this example that $f_j = C_j$ for all $j$, because each link has a single-link user. Therefore, $\mu_1, \mu_2$, and $\mu_3$ can be any values in $\mathbb{R}_+$, and $x_i = 1 - x_0$ for $1 \leq i \leq 3$. Condition (7.21) becomes:

$$\beta_0 = x_0(\mu_1 + \mu_2 + \mu_3), \quad \beta_1 = x_1\mu_1, \quad \beta_2 = x_2\mu_2, \quad \beta_3 = x_3\beta_3..$$

Eliminating the $\mu$'s and using $x_i = 1 - x_0$ yields:

$$\beta_0 = x_0 \left( \frac{\beta_1}{1 - x_0} + \frac{\beta_2}{1 - x_0} + \frac{\beta_3}{1 - x_0} \right),$$

or

$$x_0 = \frac{\beta_0}{\beta_0 + \beta_1 + \beta_2 + \beta_3}.$$

For example, if $\beta_i = 1$ for all $i$, then $x = (0.25, 0.75, 0.75, 0.75)$, and $\mu = (1.333, 1.333, 1.333)$. If $\beta_3$ were increased to two, still with $\beta_0 = \beta_1 = \beta_2 = 1$, then the new allocations and link prices would be $x = (0.2, 0.8, 0.8, 0.8)$ and $\mu = (1.25, 1.25, 2.5)$. Thus, the increase in $\beta_3$ causes an increase in the allocation to user 3 from 0.75 to 0.8. A side effect is that the rates of users 1 and 2 also increase the same amount, the rate of user 0 is decreased, the price of link 3 increases substantially, while the prices of links 1 and 2 decrease somewhat.

## 7.8 Fair allocation

The problem *SYSTEM(U,A,C)* is an example of a pure allocation problem with no costs. The allocation $x$ is constrained to be in the convex set $\{x \geq 0 : Ax \leq C\}$. A generalization is to replace the space of $x$ by a general convex set $\Lambda$. For example, we could let $\Lambda = \{x \geq 0 : \sum_{r \in R} x_r^2 \leq C\}$. This leads to the following generalization of *SYSTEM(U,A,C)*:

*ALLOCATE(U,Λ)*

$$\max \sum_{r \in R} U_r(x_r)$$

over

$$x \in \Lambda.$$

An optimal solution exists if $\Lambda$ is closed and bounded and the functions $U_r$ are continuous. By Proposition 7.1.1, $x^*$ is optimal if and only if

$$\sum_{r \in R} U_r{}'(x_r^*)(x_r - x_r^*) \leq 0 \quad \text{for all } x \in \Lambda. \tag{7.22}$$

In case $U_r(x_r) = \beta_r \log x_r$ for all $r$, for some constants $\beta_r > 0$, the optimality condition (7.22) becomes

$$\sum_{r \in R} \beta_r \left\{ \frac{x_r - x_r^*}{x_r^*} \right\} \leq 0 \text{ for all } x \in \Lambda. \tag{7.23}$$

The term in braces in (7.23) is the weighted normalized improvement of $x_r$ over $x_r^*$ for user $r$. For example, if $\frac{x_r - x_r^*}{x_r^*} = 0.5$, we say that $x_r$ offers factor 0.5 increase over $x_r^*$. The optimality condition (7.23) means that no vector $x$ can offer a positive weighted sum of normalized improvement over $x^*$. A vector $x^*$ that satisfies (7.23) is said to be a *proportionally fair* allocation, with weight vector $\beta$.

If instead, $U_i(x) = \left(\frac{\beta_i}{1-\alpha}\right)(x^{1-\alpha} - 1)$ for each $i$, where $\beta_i > 0$ for each $i$, and $\alpha \geq 0$ is the same for all users, then an optimal allocation is called $\alpha$-fair, and the optimality condition for $\alpha$-fairness is

$$\sum_{i=1}^{n} \beta_i \left\{ \frac{x_i - x_i^*}{(x_i^*)^\alpha} \right\} \leq 0 \text{ for all } x \in \Lambda.$$

The limiting case $\alpha = 1$ corresponds to proportional fairness. The case $\alpha = 0$ corresponds maximizing a weighted sum of the rates. A limiting form of $\alpha$ fairness as $\alpha \to \infty$ is max-min fairness, discussed next.

*Max-Min Fair Allocation*

In the remainder of this section we discuss max min fairness, an alternative to the notion of proportional fairness. Perhaps the easiest way to define the max-min fair allocation is to give a recipe for finding it. Suppose that $\Lambda$ is a closed, convex subset of $\mathbb{R}_+^n$ containing the origin. Let $e_r$ denote the vector in $\mathbb{R}^n$ with $r^{th}$ coordinate equal to one, and other coordinates zero. Given $x \in \Lambda$, let $g_r(x) = 1$ if $x + \epsilon e_r \in \Lambda$ for sufficiently small $\epsilon > 0$, and $g_r(x) = 0$ otherwise. Equivalently, $g(x)$ is the maximal vector with coordinates in $\{0, 1\}$ such that $x + \epsilon g(x) \in \Lambda$ for sufficiently small $\epsilon > 0$. Construct the sequence $(x^{(k)} : k \geq 0)$ as follows. Let $x^{(0)} = 0$. Given $x^{(k)}$, let $x^{(k+1)} = x^{(k)} + a_k g(x^{(k)})$, where $a_k$ is taken to be as large as possible subject to $x^{(k+1)} \in \Lambda$. For some $K \leq n$, $g(x^{(K)}) = 0$. Then $x^{(K)}$ is the max-min fair allocation. We denote it by $x^{maxmin}$.

Equivalently, if the coordinates of $x^{max-min}$ and any other $x \in \Lambda$ are both reordered to be in nondecreasing order, then the reordered version of $x^{max-min}$ is lexicographically larger than the reordered version of $x'$.

Another characterization is that for any $x \in \Lambda$, if $x_r > x_r^{max-min}$ for some $r$, then there exists $r'$ so that $x'_r < x_{r'}^{max-min} \leq x_r^{max-min}$.

Still another characterization can be given for $x^{max-min}$, in the special case that $\Lambda = \{x \geq 0 : Ax \leq C\}$, as in the congestion control problem with hard link constraints. Setting $f^{max-min} = Ax^{max-min}$, the characterization is as follows: For each route $r$ there is a link $j_r$ so that $f_j^{max-min} = C_j$ and $x_r^{max-min} = \max\{x_{r'}^{max_min} : j \in r'\}$. Link $j_r$ is called a *bottleneck* link for route $r$. In words, a bottleneck link for route $r$ must be saturated, and no other user using link $j_r$ can have a strictly larger allocation than user $r$. Due to this last characterization, a max-min fair allocation is sometimes called the *envy free* allocation, because for each user $r$ there is a bottleneck link, and user $r$ would not get a larger allocation by trading allocations with any of the other users using that link.

## 7.9 Further reading and notes

The flow deviation algorithm was introduced in [2] and is discussed in [7], and is a special case of the Frank-Wolfe method. The flow deviation algorithms and other algorithms, based on use of second derivative information and projection, for congestion control and routing, are covered in [1]. For the decomposition of system problems into user and network problems, we follow [3]. Much additional material on the primal and dual algorithms, and second order convergence analysis based on the central limit theorem, is given in [4]. The notion of $\alpha$-fairness is given in [5]. Additional material on algorithms and convergence analysis can be found in [6].

Today many subnetworks of the Internet use shortest path routing algorithms, but typically the routing is not dynamic, in that the link weights do not depend on congestion levels. Perhaps this is because the performance is deemed adequate, because there is plenty of capacity to spare. But it is also because of possible instabilities due to time lags. The recent surge in the use of wireless networks for high data rates has increased the use of dynamic routing (or load balancing) methods. Congestion control is central to the stability of the Internet, embodied by the transport control protocol (TCP) and van Jacobson's slow start method. The decomposition method discussed in this chapter matches well the congestion control mechanism used on the Internet today, and it gives insight into how performance could be improved.

## 7.10 Problems

**1. A flow deviation problem (Frank-Wolfe method)**
Consider the communication network with 24 links indicated in Figure 7.5. Each undirected

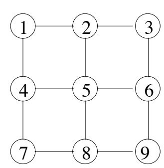



Figure 7.5: Mesh network with one source and one destination

link in the figure represents two directed links, one in each direction. There are four users: $S = \{(1,9),(3,7),(9,1),(7,3)\}$, each with demand $b$. The initial routing is deterministic, being concentrated on the four paths $(1,2,5,6,9)$,$(3,6,5,8,7)$,$(9,8,5,4,1)$, and $(7,4,5,2,3)$. Suppose that the cost associated with any one of the 24 links is given by $D_j(f) = f_j^2/2$, where $f_j$ is the total flow on the link, measured in units of traffic per second.
(a) What is the cost associated with the initial routing?
(b) Describe the flow (for all four users) after one iteration of the flow deviation algorithm. Assume that, for any given user, any route from the origin to the destination can be used. Is the resulting flow optimal?

**2. A simple routing problem in a queueing network**
Consider the open queueing network with two customer classes shown in Figure 7.6. Customers of

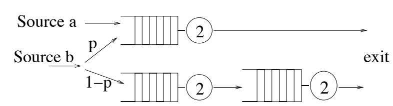



Figure 7.6: Open network with two types of customers

each type arrive according to a Poisson process with rate one. All three stations have a single exponential server with rate two, all service times are independent, and the service order is first-come, first-served. Each arrival of a customer of type $b$ is routed to the upper branch with probability $p$, independently of the history of the system up to the time of arrival. (a) Find expressions for $D_a(p)$ and $D_b(p)$, the mean time in the network for type $a$ and type $b$ customers, respectively. (b) Sketch the set of possible operating points $\{(D_a(p), D_b(p)) : 0 \leq p \leq 1\}$. (c) Find the value $p_a$ of $p$ which minimizes $D_a(p)$. (d) Find the value $p_b$ of $p$ which minimizes $D_b(p)$. (e) Find the value $p_{ave}$ of $p$ which minimizes the average delay, $(D_a(p) + D_b(p))/2$. Also, how many iterations are required for the flow deviation algorithm (Frank-Wolfe method) to find $p_{ave}$? (f) Find the value $p_m$ of $p$ which minimizes $\max\{D_a(p), D_b(p)\}$.

**3. A joint routing and congestion control problem**
Consider the network shown in Figure 7.7 and suppose there are only two users, with respective

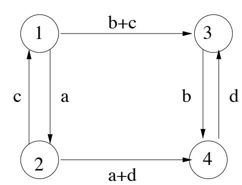



Figure 7.7: Network with joint routing and congestion control

origin destination pairs (1,4) and (2,3). Suppose that the system cost for flow $f_j$ on any link $j$ is $D_j(f_j) = \frac{f_j^2}{2}$, and that the utility of flow $x_{i,j}$ for origin-destination pair $(i, j)$ is $\beta_{ij} \log(r_{ij})$ for constants $\beta_{14}, \beta_{23} > 0$. Suppose that each user is served by two routes. For convenience of notation we use the letters $a, b, c, d$ to denote the values of the four route flows: $a = x_{124}$, $b = x_{134}$, $c = x_{213}$, and $d = x_{243}$. The joint routing and congestion control problem is to minimize the sum of link costs minus the sum of the utilities:

$$\min_{a,b,c,d\geq 0} \frac{1}{2}(a^2 + b^2 + c^2 + d^2 + (b + c)^2 + (a + c)^2) - \beta_{14} \log(a + b) - \beta_{23} \log(c + d)$$

(a) Write out the optimality conditions. Include the possibility that some flow values may be zero. (b) Show that if all flow values are positive, then $a = b$ and $c = d$. (c) Find the optimal flows if $\beta_{14} = 66$ and $\beta_{23} = 130$. Verify that for each route used, the price of the route (sum of $D_j'(f_j)$ along the route) is equal to the marginal utility of flow for the origin-destination of the route.

**4. Sufficiency of the optimality condition for hard link constraints**
The goal is to prove that if $x^*, y^*$ satisfy the constraints for the joint congestion control and routing

problem with hard link constraints, *SYSTEM(U,H,A,C)*, and if together with some $\mu$ they satisfy (7.9) and (7.10), then $x^*, y^*$ is a solution to the problem.
(a) By maintaining $x = Hy$, we consider $y$ to be the only independent variable. Define the Lagrangian

$$L(y, \mu) = \sum_s U_s(x_s) + \sum_j \mu_j \left( C_j - \sum_{r:j \in r} y_r \right)$$

Show that $L(y^*, \mu) \geq L(y, \mu)$ for any other vector $y$ with $y \geq 0$. (Hint: There is no capacity constraint, and $L(y, \mu)$ is concave in $y$). (b) Deduce from (a) that $x^*, y^*$ is a solution to the system optimization problem. (Note: Since the feasible set is compact and the objective function is continuous, a maximizer $y^*$ of $L(y, \mu)$ subject to $Ay \leq C$ over $y \geq 0$ exists. Since the objective function is continuously differentiable and concave and the constraints linear, a price vector $\mu$ satsisfying (7.9) and (7.10) also exists, by a standard result in nonlinear optimization theory based on Farkas' lemma.)

**5. Fair flow allocation with hard constrained links**

Consider four flows in a network of three links as shown in Figure 7.8. Assume the capacity of

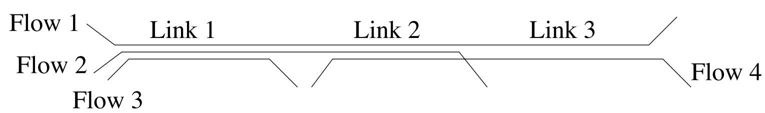


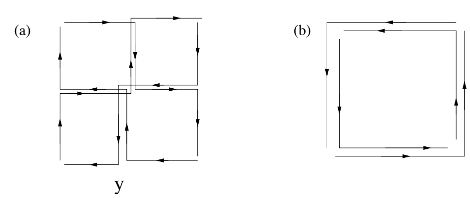

Figure 7.8: Network with four flows

each link is one. (a) Find the max-min fair allocation of flows. (b) Find the proportionally fair allocation of flows, assuming the flows are equally weighted. This allocation maximizes $\sum_r \log(x_r)$, where $x_r$ is the rate of the $r$th flow. Indicate the corresponding link prices, $\mu_j$.

## 7.11 Solutions

**1. A flow deviation problem (Frank-Wolfe method)**

(a) Initially the 8 links directed counter clockwise along the perimeter of the network carry zero flow, and the other 16 links carry flow $b$, as shown in part (a) of the figure. So the initial cost is $F(y) = 8b^2$.

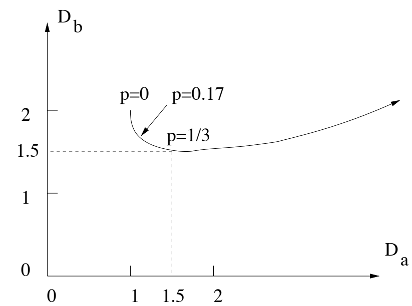



(b) Each of the four minimum first derivative length paths $r_s^*$ use four of the eight links not used initially, as shown in part (b) of the figure. Thus, $\text{cost}(\overline{y}(\alpha)) = 8(2\alpha b)^2 + 16((1-\alpha)b)^2$. Minimizing over $\alpha$, we find $\alpha^* = \frac{1}{3}$. Under the new flow, all 24 links carry flow $\frac{2b}{3}$. The first derivative link length of all paths used is minimum. So $\overline{y}$ is an optimal flow.

**2. A simple routing problem in a queueing network**
In equilibrium, the stations are independent and have the distribution of isolated M/M/1 queues, and the delay in an $M/M/1$ queue with arrival rate $\lambda$ and departure rate $\mu$ is $\frac{1}{\mu-\lambda}$. Therefore,

$$D_a = \frac{1}{2-(1+p)} \quad \text{and} \quad D_b(p) = p\left(\frac{1}{2-(1+p)}\right) + (1-p)\left(\frac{1}{2-(1-p)} + \frac{1}{2-(1-p)}\right)$$

(b)




(c) $p = 0$ minimizes $D_a$, because $D_a$ is increasing in $p$.
(d) Setting $D_b'(p) = 0$ yields $p = \frac{1}{3}$.
(e) Setting $D_a'(p) + D_b'(p) = 0$ yields $p \approx 0.17$. Only one iteration of the flow deviation algorithm is necessary because the only degree of freedom is the split for flow $b$ between two paths. The line search for flow deviation solves the problem in one iteration.
(f) The solution is $p = \frac{1}{3}$, because $D_a$ is increasing in $p$, $D_b$ is decreasing in $p$, and $D_a(\frac{1}{3}) = D_b(\frac{1}{3})$. (It is a coincidence that the answers to (d) and (f) are the same.)

**3. A joint routing and congestion control problem**
(a) Since $D_j'(f_j) = f_j$, the price for a link is equal to the flow on the link. For example, path 124 has price $2a + d$. The optimality conditions are:

$$\begin{aligned}
2a + d &\geq \frac{\beta_{14}}{a+b} \quad \text{with equality if } a > 0 \\
2b + c &\geq \frac{\beta_{14}}{a+b} \quad \text{with equality if } b > 0 \\
2c + b &\geq \frac{\beta_{23}}{c+d} \quad \text{with equality if } c > 0 \\
a + 2d &\geq \frac{\beta_{23}}{c+d} \quad \text{with equality if } d > 0
\end{aligned}$$

The first two conditions are for user $s = (1,2)$ and the last two are for user $s = (2,3)$.
(b) If $a, b, c, d > 0$ the equality constraints hold in all four conditions. The first two conditions yield $2a+d = 2b+c$ or, equivalently, $2a - 2b = c - d$. The last two yield $2c+b = a+2d$ or, equivalently, $2a - 2b = 4c - 4d$. Thus, $c - d = 4c - 4d$ or, equivalently, $c = d$, and hence also $a = b$. (c) We seek a solution such that all four flows are nonzero, so that part (b) applies. Replacing $b$ by $a$ and $d$ by $c$, and using $\beta_{14} = 66$ and $\beta_{23} = 130$, the optimality conditions become

$$\begin{aligned}
2a + c &= \frac{66}{2a} \\
2c + a &= \frac{130}{2c}
\end{aligned}$$

or $a = b = 3$ and $c = d = 5$. To double check, note that the two routes for $s = (1,4)$ have price 11 each, which is also the marginal value for route $(1,4)$, and the two routes for $s = (2,3)$ have price 13 each, which is also the marginal value for route $(2,3)$.

**4. Sufficiency of the optimality condition for hard link constraints**

Since $L(y,\mu)$ is concave in $y$, for any $\epsilon$ with $0 < \epsilon < 1$,

$$L(y,\mu) - L(y^*,\mu) \leq \frac{1}{\epsilon}\left\{L(y^* + \epsilon(y - y^*),\mu) - L(y^*,\mu)\right\}$$

Taking the limit as $\epsilon \to 0$ yields

$$\begin{aligned} L(y,\mu) - L(y^*,\mu) \quad &\leq \quad \sum_{s\in S}\sum_{r\in s} U_s'(x_s^*)(y_r - y_r^*) \;-\; \sum_j \mu_j\big(\sum_{r:j\in r} y_r - y_r^*\big) \\ &= \quad \sum_{s\in S}\sum_{r\in s}(y_r - y_r^*)\left\{U_s'(x_s^*) - \sum_{j\in r}\mu_j\right\} \leq 0 \qquad (7.24) \end{aligned}$$

where the final inequality follows from the fact that for each $r$, the quantity in braces in (7.24) is less than or equal to zero, with equality if $y_r^* > 0$.

(b) Note that $\sum_{j\in J}\mu_j(C_j - \sum_{r:j\in r} y_r^*) = 0$ and if $Ay \leq C$, then $\sum_{j\in J}\mu_j(C_j - \sum_{r:j\in r} y_r) \geq 0$. So if $y \geq 0$ and $Ay \leq C$,

$$\sum_{s\in S} U(x_s^*) = L(y^*,\mu) \geq L(y,\mu) \geq \sum_{s\in S} U(x_s).$$

**5. Fair flow allocation with hard constrained links**

(a) By inspection, $x_{max-min} = (\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})$.

(b) (proportional fairness) Seek a solution to the equations

$$\begin{aligned} x_1 = \frac{1}{\mu_1+\mu_2+\mu_3} \quad x_2 &= \frac{1}{\mu_1+\mu_2} \quad x_3 = \frac{1}{\mu_1} \quad x_4 = \frac{1}{\mu_2+\mu_3} \\ x_1 + x_2 + x_3 &\leq 1, \text{ with equality if } \mu_1 > 0 \\ x_1 + x_2 + x_4 &\leq 1, \text{ with equality if } \mu_2 > 0 \\ x_1 + x_4 &\leq 1, \text{ with equality if } \mu_3 > 0 \end{aligned}$$

Clearly $x_1 + x_4 < 1$, so that $\mu_3 = 0$. Also, links 1 and 2 will be full, so that $x_3 = x_4$. But $x_3 = \frac{1}{\mu_1}$ and $x_4 = \frac{1}{\mu_3}$, so that $\mu_1 = \mu_2$. Finally, use $\frac{1}{2\mu_1} + \frac{1}{2\mu_1} + \frac{1}{\mu_1}$ to get $\mu_1 = \mu_2 = 2$, yielding $x_{pf} = (\frac{1}{4},\frac{1}{4},\frac{1}{2},\frac{1}{2})$.

Flows 1 and 2 use paths with price $\mu_1 + \mu_2 = 4$ and each have rate $\frac{1}{4}$.

Flows 3 and 4 use paths with price $\mu_1 = \mu_2 = 2$ and each have rate $\frac{1}{2}$.

# Bibliography


[1] D. Bertsekas and R.G. Gallager, *Data Networks, 2nd ed.*, Prentice-Hall, Englewood Cliffs,New Jersey, 1992.

[2] L. Fratta, M. Gerla, and L. Kleinrock, “The flow deviation method: an approach to store and forward communication network design.” *Networks*, vol. 3, pp. 97-133, 1973.

[3] F. P. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications,* Vol. 8, (1997), pp. 33-37.

[4] F. P. Kelly, A.K. Maulloo, and D.K.H. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society* 49 (1998), 237-252.

[5] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Trans. Networking*, vol. 8, pp. 556-567, 2000.

[6] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhäuser, Boston, 2003.

[7] L. Kleinrock,, *Queueing Systems, Vol, 2: Computer Applications*, Wiley, 1976.