Lec25-mwf

Monday, March 23, 2009 9:42 PM

Solution for the sample exam on the web.

- The capacity region (with time-sharing) becomes

$$\sum_{k=1}^{M} \theta_k \, \mathrm{Conv}\{\vec{r} \mid \vec{r} = g(\vec{P}, S_k), \vec{P} \in \Pi\}$$

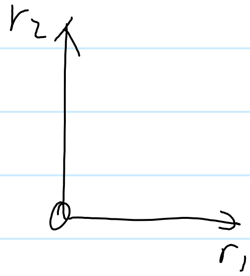
<u>Example:</u>

- One BS
- Two users
- BS can only transmit to one user at a time
- Four states

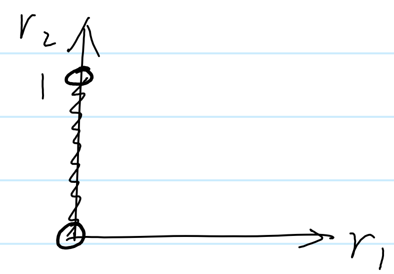
(0, 0) (0, 1) (1, 0) (1, 1)

ON for user 1 OFF for user 2

- The convex-hull in each state

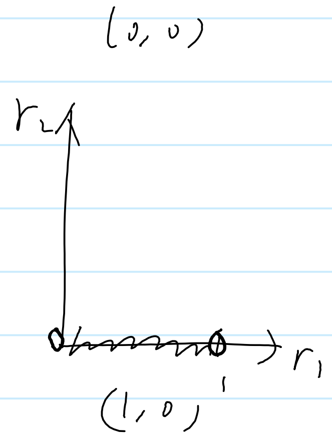


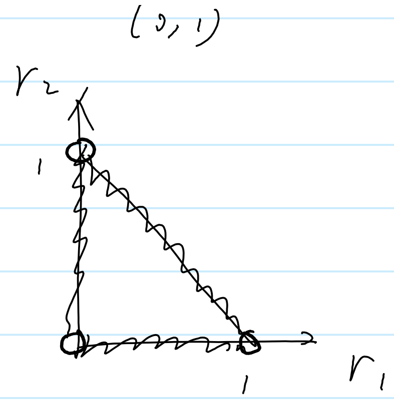
(0, 0)



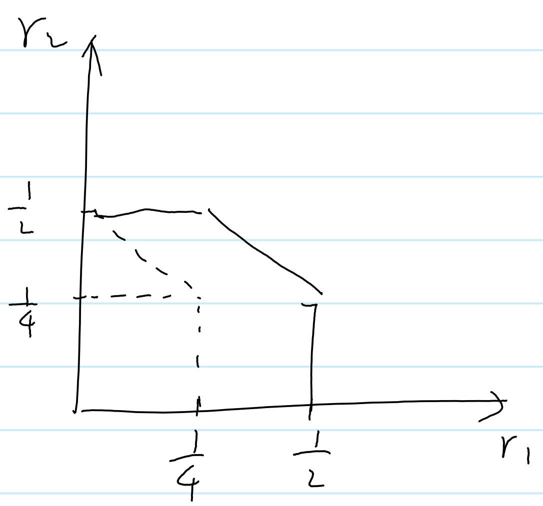
(0, 1)

r_2,





- Suppose $p_{00} = p_{01} = p_{10} = p_{11} = \frac{1}{4}$.
- What does the overall capacity region looks like.



Joint congestion control and scheduling

Friday, March 27, 2009 10:59 AM

- Consider the following convex problem.

$$\max_{\vec{x}} \sum_s U_s(x_s)$$

$$\text{sub to} \quad \sum_s H_s^l x_s \le r_l \quad \forall l \quad (*)$$

$$\vec{r} \in \text{Conv}\{\vec{r} \,/\, \vec{r} = g(\vec{p}),\ \vec{p} \in \Pi\}$$

- fixed routing
- no channel variation

- $U_s(\cdot)$: utility / fairness
- H_s^l : fixed routing
- $\vec{r}$: not given yet.
- No channel variation (for simplicity)

- Associate a Lagrange multiplier q_l for the constraint $(*)$
  - do not use a Lagrange multipler for the last constraint.

$$L(\vec{x}, \vec{r}, \vec{q}) = -\sum_s U_s(x_s) + \sum_l q_l \left(\sum_s H_s^l x_s - r_l\right)$$

$$= -\sum_s \left[U_s(x_s) - x_s \cdot \sum_l H_s^l q_l\right] - \sum_l q_l r_l$$

- To minimize the Lagrangian,
  - each user should choose x_s by

$$\max_{x_s} \ U_s(x_s) - x_s \cdot \sum_l H_s^l q_l$$

- the vector $\vec{r}$ should be chosen by

$$\max_{\vec{r} \in \mathrm{Conv}\{g(\vec{p}) \mid \vec{p} \in \Pi\}} \sum_l q_l r_l$$

- Again, we can view q_l as the "price" of resources at link l

  - each user maximizes the net utility

  - "max-weight scheduling": the schedule $\vec{r}$ is chosen to maximize the overall value of the available resource

- The subgradient of the dual is again given by

$$\sum H_l^s x_s(t) - r_l(t)$$

- Hence, the dual variables can be updated by

$$q_l(t+1) = \Big[q_l(t) + \gamma \Big(\sum_s H_l^s \underbrace{x_s(t)}_{\text{max the net utility}} - \underbrace{r_l(t)}_{\text{max weight scheduling}} \Big) \Big]^+$$

- As $t \to +\infty$, $q_l(t) \to q_l^*$ (or a neighborhood around q_l^*) when γ is appropriately chosen.

Ref:
X. Lin and N. B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks," in

Proceedings of the IEEE Conference on Decision and Control, Paradise Island, Bahamas, December 2004.

35

Max-weight scheduling

Friday, March 27, 2009 11:10 AM

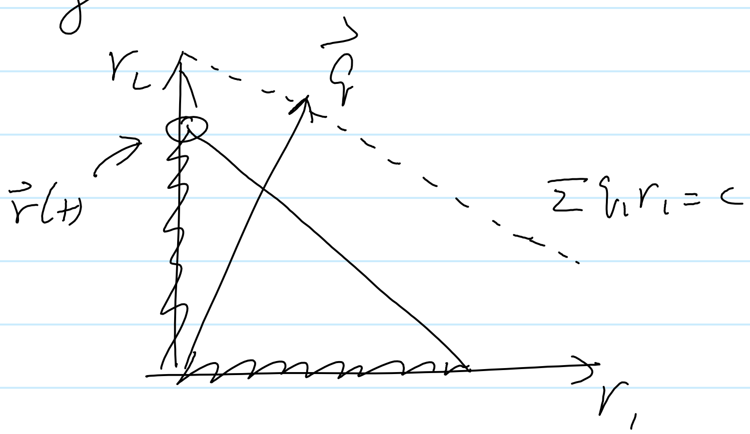
No need for the convex hull

$$\max_{\substack{\vec{r} \in \mathrm{Conv}\{g(\vec{p});\\ \vec{p} \in \Pi}} \sum_l q_l r_l = \max_{\substack{\vec{r} = g(\vec{p})\\ \vec{p} \in \Pi}} \sum_l q_l r_l$$

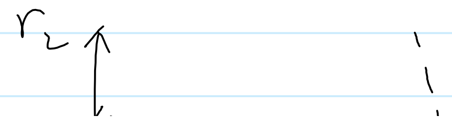
Non-Convergence

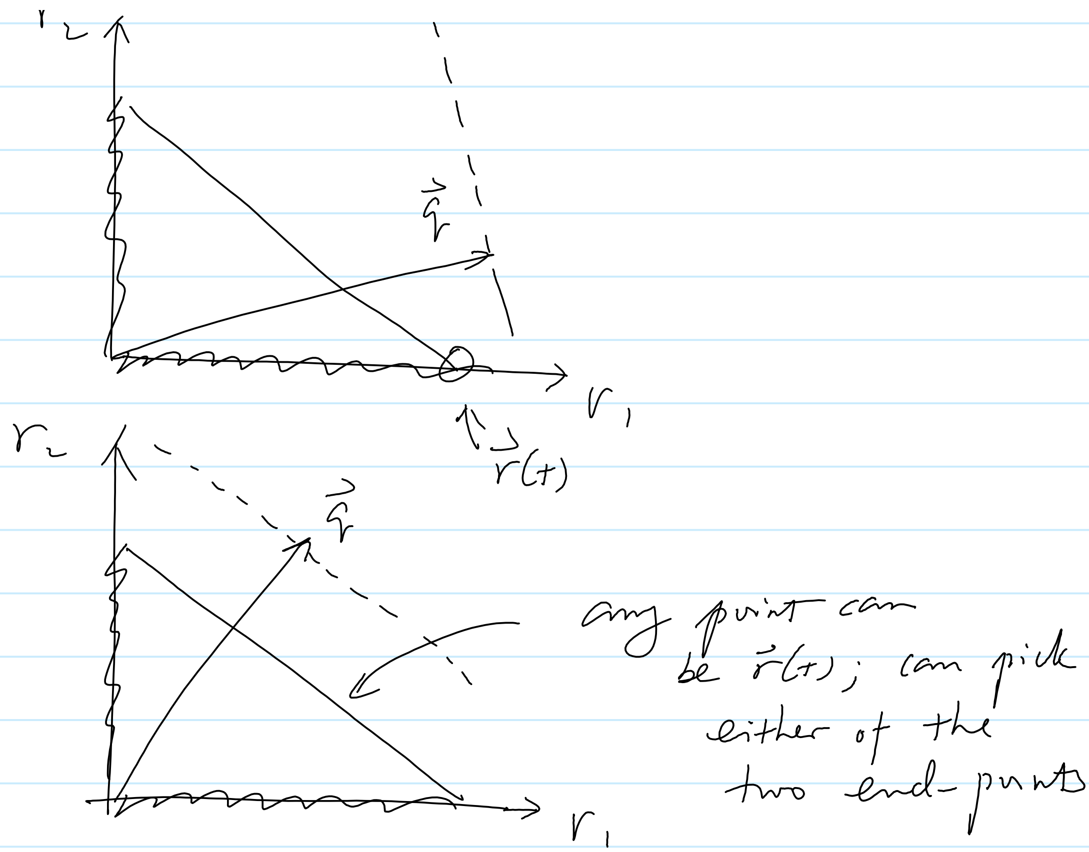
- As $q_l(t) \to q_l^*$, the rates x_s should also converge to some optimal value x_s^* by the KKT condition. (Assume that the utility functions are strictly concave)

- However, in general the value of $\vec{r}(t)$ from max-weight scheduling does not converge



capacity region of two-users (no channel variations)





- In general, we won't have

$$\sum_s H_l^s x_s \le r_l(t) \quad \forall l$$

at most iterations.

(Q) In what sense, does the max-weight scheduling solve the original problem?

(A) in a time-averaging sense

- Recall that

$$q_l(t+1) = \left[q_l(t) + \gamma \left(\sum_s H_l^s x_s(t) - r_l(t) \right) \right]^+$$

$$q_l(t+1) = \left[q_l(t) + \gamma\left(\sum_s H_{ls}\, x_s(t) - r_l(t)\right)\right]$$

$$\Rightarrow \quad q_l(t+T) \geq q_l(t) + \gamma\left[\sum_s H_{ls}^{l} \sum_{h=t}^{t+T} x_s(h) - \sum_{h=t}^{t+T} r_l(t)\right]$$

$$\Rightarrow \quad \sum_s H_s^{l}\left(\frac{1}{T}\sum_{h=t}^{t+T} x_s(h)\right) \leq \frac{1}{T}\sum_{h=t}^{t+T} r_l(h) + \frac{q_l(t+T) - q_l(t)}{\gamma T}.$$

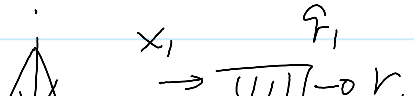
- Since $\vec{q}(t)$ converges (and hence is bounded), the last term $\to 0$ as T becomes large

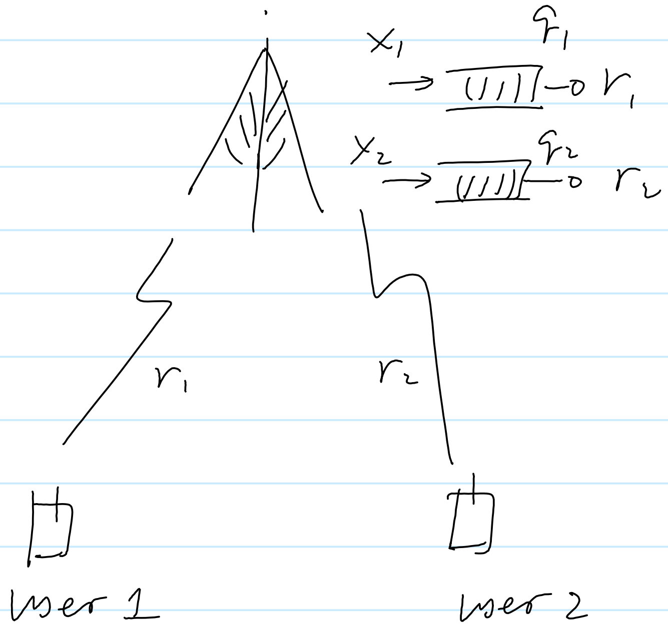
- Also, as $t \to +\infty$, $x_s(t) \to x_s^*$

- Hence, in the long-run (i.e. for t, T sufficiently large,

$$\sum_s H_s^{l} x_s^* \leq \frac{1}{T}\sum_{h=t}^{t+T} r_l(h) \pm \varepsilon.$$

$\Rightarrow$ The max-weight scheduling policy finds the right combination of vectors $\vec{r}$ and their fractions of time to support the incoming rate.





50

Add routing

Friday, March 27, 2009 11:38 AM

Model

- x_s : rate of user s
- f_s, d_s : source / destination of user s
- r_{ij}^s : amount of capacity on link (i,j) that is allocated for flow s.

$$\vec{r} = [r_{ij}^s, (i,j) \in L,]$$

Problem formulation

$$\max \quad \sum_s U_s(x_s)$$

$$\text{sub to} \quad \sum_{j:(i,j)\in L} r_{ij}^s - \sum_{j:(j,i)\in L} r_{ji}^s - \sum_{s: f_s = i} x_s \geq 0$$

$\forall$ all nodes i & all flows s such that $d_s \neq i$

$$\left[\sum_s r_{ij}^s\right] = r_{ij}$$

$$[r_{ij}] \leq g(\vec{P})$$

rate-power function

power-assignment

- However, as we discuss before, $g(\cdot)$ usually

- However, as we discuss before, $g(\cdot)$ usually does not lead to a convex constraint.

- Using the time-sharing idea, we can replace the last constraint by

$$[r_{ij}] \in \mathrm{Conv}\{\vec{r} = g(\vec{P});\ P \in \Pi\}$$

- This is now a convex problem!

Solution:

- Associate a Lagrange multiplier q_i^s for the node-balance constraint

$$L(\vec{x}, \vec{r}, \vec{q}) = -\sum_s U_s(x_s)$$

$$+ \sum_{i,s:\, d_s \neq i} q_i^s \Big[\sum_{j:(j,i)\in L} r_{ji}^s + \sum_{s: f_s = i} x_s$$

$$- \sum_{j:(i,j)\in L} r_{ij}^s\Big]$$

$$= -\sum_s \left[U_s(x_s) - q_{f_s}^s x_s\right]$$

$$- \sum_{\substack{i,s:\\ d_s \neq i}} q_i^s \Big[\sum_{j:(i,j)\in L} r_{ij}^s - \sum_{j:(j,i)\in L} r_{ji}^s\Big]$$

Hence, to minimize the Lagrangian over $\vec{x}, \vec{r}$

- Each user should maximize

$$U_s(x_s) - q_{f_s}^s x_s$$

  - Maximize net utility where the price is given by the length of the first queue

- The schedule $\vec{r}(t)$ should be chosen to maximize

$$\sum_{i,s} q_i^s \left[\sum_{j:(i,j)\in L} r_{ij}^s - \sum_{j:(j,i)\in L} r_{ji}^s \right]$$

$$= \sum_{(i,j)\in L} \sum_s \left(q_i^s - q_j^s \right) r_{ij}^s$$

  - $q_{d_s}^s = 0$

  - over the constraint

$$\left[\sum_s r_{ij}^s \right] = r_{ij} \qquad [r_{ij}] \in \text{Conv}\{\vec{r} = g(\vec{p}) ; p \in \Pi\}$$

- The dual variables can be updated by

$$q_i^s(t+1) = \left[q_i^s(t) + \gamma \left(\sum_{j:(j,i)\in L} r_{ji}^s(t) + \sum_{s: f_s = i} x_s - \sum_{j:(i,j)\in L} r_{ij}^s(t) \right) \right]^+.$$

- Note that q_i^s can again be interpreted as the backlog of packets at node i that are from flow s.

Backpressure routing and max-weight scheduling: random

Tuesday, March 31, 2009 4:07 PM

- The scheduling component corresponds to the so called back-pressure routing

$$\max_{[r_{ij}^s]} \sum_{(i,j)\in L} \sum_s (\tilde{q}_i^s - \tilde{q}_j^s) r_{ij}^s$$

sub to $\quad r_{ij}^s \geq 0$

$$\left[\sum r_{ij}^s\right] = r_{ij}$$

$$[r_{ij}] \in \text{Conv}\{\vec{r} = g(\vec{p}); \vec{p} \in \Pi\}$$

- The difference $\tilde{q}_i^s - \tilde{q}_j^s$ is called a differential backlog
  - Recall that $\tilde{q}_i^s$ can be interpreted as scalar multiple of the backlog.

- Clearly, for a fixed sum $\sum r_{ij}^s = r_{ij}$, there is no reason to use $r_{ij}^s > 0$ unless

$$\tilde{q}_i^s - \tilde{q}_j^s \geq 0 \quad \text{and}$$

$$\tilde{q}_i^s - \tilde{q}_j^s \geq \max_{s'} \tilde{q}_i^{s'} - \tilde{q}_j^{s'}$$

- Hence, on each link (i,j), we only serve the packets that correspond to the flow with the largest differential backlog ≥ 0

- This decision determines how packets are routed.

- Then, the scheduling problem becomes

$$\max \sum_{(i,j)\in L} r_{ij} \max_s (q_i^s - q_j^s, 0)$$

$$\text{sub to} \quad [r_{ij}] \in \text{Conv}\{\vec{r} = g(\cdot);\ \vec{p} \in \Pi\}$$

- This is again max-weight scheduling!

- However, since the objective is linear, the solution must lie at one of the extreme points (without the convex-hull).

$$\max \sum_{(i,j)\in L} r_{ij} \max_s (q_i^s - q_j^s, 0)$$

$$\text{sub to} \quad \vec{r} \leq g(\vec{p})\ ;\ \vec{p} \in \Pi.$$

- When $g(\cdot)$ is concave, this step still corresponds to a non-convex problem
  $\Rightarrow$ may seek sub-optimal solutions.

corresponds to a non-convex problem
$\Rightarrow$ may seek sub-optimal solutions.

Note that, still, even if the dual variables converge, the routing & the scheduling decision does not converge

- Packets of multiple flows will be served in a time-interleaved fashion
- Links will be turned ON/OFF (or assigned different powers) in a time-interleaved fashion

The rates x_s do converge (provided that the Utility functions are strictly convex.

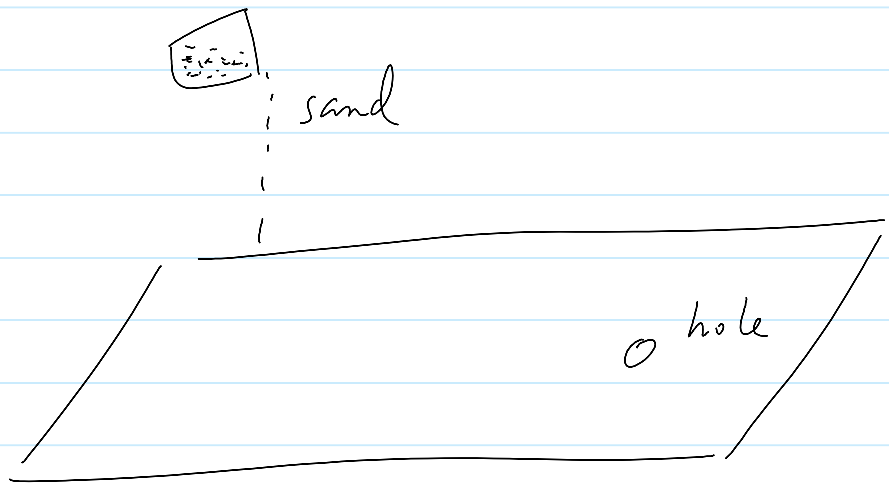
The primal variables r_{ij}^s's are optimal in the sense that the node-balance equations are satisfied by the time-averaged values.

An analogy-skip

Tuesday, March 31, 2009 4:18 PM

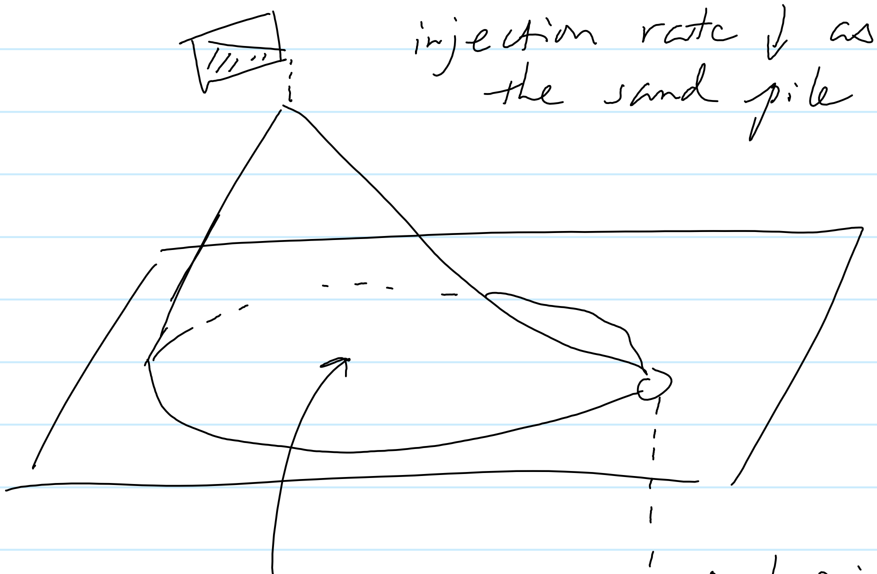
- With a single flow

Problem:



Solution:

injection rate ↓ as the sand pile builds up



sand particle flows along the direction with the positive backlog

out-going rate = incoming rate.

(= routing)

75