Optimization in Communication Networks Lecture 4: Network Utility Maximization

Prof. Yung Yi, yiyung@ee.kaist.ac.kr,
`http://lanada.kaist.ac.kr`
EE, KAIST

March 13, 2016

Lecture Outline

- Application of Convex Optimization, especially Lagrange Duality

- Internet Congestion Control

- α-Fair Allocation

- Readings
  - $\star$Dual-based congestion control: [Low and Lapsley, 1999]
  - $\star\alpha$-fairness: [Mo and Walrand, 2000]
  - The seminal paper on NUM: [Kelly et al., 1998]
  - Duality model of TCP: [Low, 2003]
  - Chapter 9.2 and 9.3 by Boyd for basics on gradient algorithms.
  - "Communication Networking: An Analytical Approach" by Anurag Kumar, pages 408 - 421.
  - "Rate adaptation, Congestion Control and Fairness: A Tutorial" by Jean-Yves Le Boudec. Sections 1.1 and 1.2. (This will be helpful if you read the entire tutorial).

Lecture Outline

- How to share network resources among flows?

- Internet congestion control (e.g., TCP): distributed algorithm to share network resource efficienctly

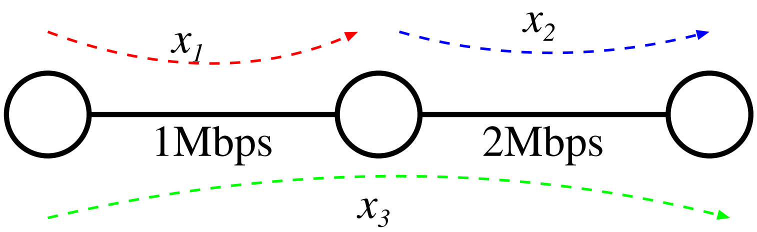
- How can we understand Internet congestion control theoretically? Does TCP have provable performance in this mind set? One answer: use optimization theory

- Internet congestion control → network resource allocation problem → network utility maximization (NUM)

Resource Sharing

- Two important metrics for "good" resource sharing: efficiency and fairness
  - Efficiency: no waste of resources, Is the resulting allocation on the boundary of the constraint set?
  - Fairness: choosing one point on the boundary of the constraint set
    cf) No general agreement on "fairness"

- Example in the class: 3 nodes with 2 links



- One nice candiate for systematic study: NUM (Network Utility Maximization)

Efficiency and Fairness

Efficiency: Pareto Efficiency

- Concept: No waste of resources

- For two n-dimensional vectors $x', x \in \mathcal{X}$ for some $\mathcal{X}$, a vector $x' = (x'_n)$ *Pareto dominates* $x = (x_n)$ if: $x'_n \geq x_n$ for all n, and the inequality is strict for at least one n.

- A vector x is Pareto efficient if it is not Pareto dominated by any other $x' \in X$.

- Can't make one player better off without making another worse off (Understand this if z is not Pareto efficient)

- Understand using a simple example in the class.

Fairness

- Difficult to define: subjective views

- However, there will be several notions of fairness definitions that many people may agree to.

  Max-min, proportional-fair, etc.

- Three questions?

  Q1. How to define fairness under what philosophy?

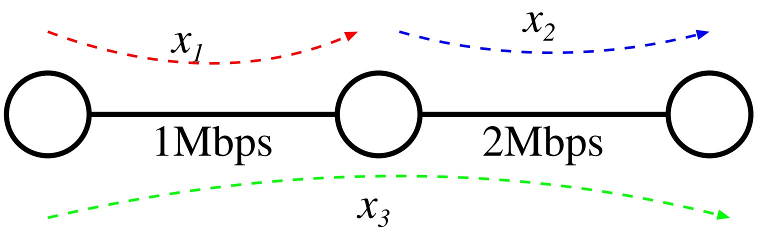
  Q2. For a given fairness definition, how to numerically compute a fairness point (i.e., centralized computation and interesting to network designers)

  Q3. How to develop a distributed algorithm that generates a fairness point (i.e., distributed computation and interesting to network operators)

  Example question: Is TCP in the Internet fair?

Max-Min Fair

- Definition. A feasible rate vector x is max-min fair if it is not possible to increase the rate of a session s, *while maintaining feasibility*, without reducing the rate of some session s' with $x_{s'} \leq x_s$.



Is (1,2,0) Pareto-efficient and/or max-min fair?

Is (0.5,1.5,0.5) Pareto-efficient and/or max-min fair?

- Important property

- P1. Look at the smallest rate in a max-min fair rate vector. It has the largest value of the minimum rate.

- P2. Among all feasible rate vectors with this value of the minimum rate, consider the next larger rate. The max-min fair rate vector has the largest value of the next larger rate as well, and so on.

Max-Min Fair: Centralized Algorithm

Step 1. The rates of all flows are increased at the same pace until one or more links are saturated
Then, the rate of flows passing over saturated links are then frozen, and the other flows continue to increase rates

Step 2. Repeat Step 1 until all rates are frozen

Check this for the example in the previous page.

Proportional Fairness and α-fairness

- Definition. (Proportional-fair) A rate vector x is proportionally fair if it is feasible and for any other feasible rate vector y, the aggregate of proportional change is negative, i.e.,

$$\sum_i \frac{y_i - x_i}{x_i} \leq 0.$$

Is (0.5,1.5,0.5) PF?

- Definition. ((w, α)-proportional-fair) A rate vector x is proportionally fair if it is feasible and for any other feasible rate vector y,

$$\sum_i w_i \Big(\frac{y_i - x_i}{x_i^{\alpha}} \Big) \leq 0.$$

PF: Central Computation

- From the definition, the PF rate vector x is such that for all other feasible y, we have:

$$\nabla U(x)(y - x) \leq 0, \quad \text{where} \quad U(x) = \sum_i \log(x_i).$$

- Thus, the PF vector x maximizes $\sum_i \log(x_i)$.

- That's what you have solved in the homework.

- This gives some motivation to study fairness from the perspective of utility function.

- Formally, we call this Network Utility Maximization.

Network Utility Maximization

Network Utility Maximization

Basic problem formulation:

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & Rx \preceq c \\ & x \succeq 0 \end{array}$$

- Objective: total utility (each U_s is smooth, increasing, concave)

  Why concave is reasonable?

- Constraint: linear flow constraint

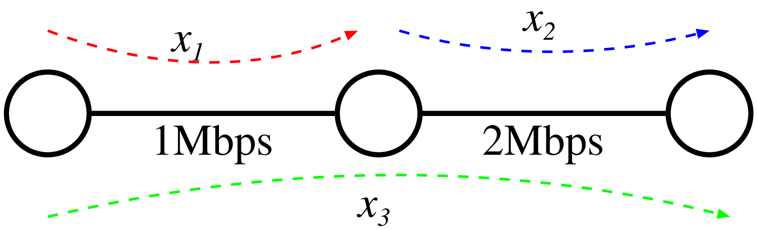
- s index of sources and l index of links

- Given routing matrix R_{ls}: 1 if flow from source s uses link l, 0 otherwise

- x_s: source rate (variables)

- c_l: link capacity (constants)

Example



- What is R?

- What is c?

Philosophy

- Maximization: pursue efficiency
- Shape of utility function: pursue fairness

  It would be good to cover many "conventional" fairness by selecting appropriate utility functions
- Questions
  - Practicability: how can we solve NUM in a distributed manner?
    Distributed optimization
  - Which utility functions → which fairness? How to prove?
    Max-min fair, Proportional fair, α-fair
  - Can TCP (developed from engineering heuristics) be reverse-engineered using NUM framework? see [Low, 2003] (not covered in this class)

Dual-based Distributed Algorithm

Extension of network flow problem, many applications

Convex optimization with zero duality gap
Lagrangian decomposition ($(p_l)_{l \in L}$ are Lagrange multipliers):

$$
\begin{aligned}
L(x,p) &= \sum_s U_s(x_s) + \sum_l p_l \left(c_l - \sum_{s:l \in L(s)} x_s \right) \\
&= \sum_s \left[U_s(x_s) - \left(\sum_{l \in L(s)} p_l \right) x_s \right] + \sum_l c_l p_l \\
&= \sum_s L_s(x_s, q_s) + \sum_l c_l p_l
\end{aligned}
$$

Dual function: $\max_{x \geq 0} L(x,p)$, and let's denote by $x^\star(p) = \arg\max_{x \geq 0} L(x,p)$.
Dual problem:

$$
\begin{array}{ll}
\text{minimize} & g(p) = L(x^*(p), p) \\
\text{subject to} & p \succeq 0
\end{array}
$$

Source algorithm: Solving dual function
Link algorithm: Solving dual problem.

Algorithm solving an optimization problem: Gradient algorithm

- There are many "gradient" based algorithms. Here is just a very short review

- $\min f(x)$, where $f : \mathcal{R}^n \mapsto \mathcal{R}$ and $f \in C^1$.

- Gradient type algorithms:

$$x^{k+1} = x^k + \alpha^k d^k, \quad k = 0, 1, 2, \ldots$$

- Choose $d^k = -\nabla f(x^k)$.

- Lemma. Any direction d that satisfies $\nabla f(x)d < 0$ is a descent direction of f at x. That is, let $x_\alpha = x + \alpha d$ where $\nabla d < 0$. Then $\exists \bar{\alpha} > 0$, such that for all $\alpha \in (0, \bar{\alpha}]$, $f(x_\alpha) < f(x)$.

- Subgradient method: used when the f is not differentiable

- Note that the above is for unconstrained optimizations. But, our problem is a constrained optimization, so we have to pick x^k that is *projected onto* the constraint set, i.e., $p \geq 0$.

Dual-based Distributed Algorithm

- Source algorithm:

$$x_s^*(q_s) = \text{argmax}\left[U_s(x_s) - q_s x_s\right], \quad \forall s$$

  Selfish net utility maximization locally at source s

- Link algorithm (gradient or subgradient based):

$$p_l(t{+}1) = \left[p_l(t) - \alpha(t)\left(c_l - \sum_{s:l\in L(s)} x_s^*(q_s(t))\right)\right]^+, \quad \forall l$$

- Balancing supply and demand through pricing

- Certain choices of step sizes $\alpha(t)$ ($e.g.$, $\alpha(t) = 1/t$) of distributed algorithm guarantee convergence to globally optimal (x^*, p^*)

Achieving fairness via NUM

- α-fair allocation

  When $\alpha \neq 1$,

  $$U(x) = \frac{x^{1-\alpha}}{1-\alpha}$$

  When $\alpha = 1$,

  $$U(x) = \log(x).$$

- In particular, α-fair allocation defines well-known notions of fairness
  - $\alpha = 1$: Proportional fairness
  - $\alpha \to \infty$: max-min fairness
  - $\alpha \to 0$: max-throughput (sum rate maximization)

- How to prove achieving max-min fairness when $\alpha \to \infty$?

- Read Section III of [Mo and Walrand, 2000]

Summary

- A nice application of Lagarange duality
- Naturally distributed algorithms, nice!
- Actually, TCP has been reverse engineered in this framework
- The choice of utility function determines network fairness
- Assumptions
  - Routes are fixed
  - Wired networks (no interference among link transmissions)
  - Only elastic data (concave utility functions)
  - No detailed QoS (e.g., minimum rate guarantee)
  - Cannot model other performance metric such as delay
  - Each session is infinitely backlogged
- Important to know what we want to model, and model it!

References

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

[Low, 2003] Low, S. H. (2003). A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536.

[Low and Lapsley, 1999] Low, S. H. and Lapsley, D. E. (1999). Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, pages 861–875.

[Mo and Walrand, 2000] Mo, J. and Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567.