ELE539A: Optimization of Communication Systems
Lecture 13: Network Utility Maximization

Professor M. Chiang
Electrical Engineering Department, Princeton University

February 28, 2005
Lecture Outline

- Resource allocation problems
- Network utility maximization
- Dual decomposition and canonical distributed algorithm
- Kelly’s decomposition and primal algorithm
Well-Known: Network Flow Problem

Minimize a linear cost subject to linear flow constraints

Classical linear programming based methods

Includes many special cases:
- Multicommodity flow problems (max flow, min cut...)
- Transportation and assignment problems
- Shortest path and spanning tree problems

Limitations in modelling nonlinearity in Internet and wireless networks
Rate Allocation Problem

Elastic traffic: applications modify their data transfer rates according to available bandwidth in communication networks

- TCP traffic over Internet TCP/IP suite
- ABR traffic over ATM networks

Q: How to share available bandwidth among competing flows of elastic traffic?

Mathematical model of rate control and distributed algorithms to understand

- Equilibrium properties: efficiency and fairness
- Dynamic properties: local and global stability
Resource Allocation Problem

Application-level view: what’s user utility?
Utility as function of QoS parameters: throughput, latency, jittering, distortion, energy efficiency...

Objective: maximize sum of user utilities

Network constraints:
- Link capacity, total power...
- Medium access possibilities
- Routing possibilities

Constrained nonlinear optimization problem formulations
Basic Model

Communication networks with $L$ links, each with capacity $c_l$

Sources (end user): $S$ of them, each emitting a flow with transmission rate $x_s$

Fixed single-path routing: source $s$ uses links $l \in L(s)$

Routing matrix $R$: $0-1$ matrix with $R_{ls} = 1$ iff $l \in L(s)$

Linear flow constraint on rates: $Rx \leq c$

Each source has utility $U_s(x_s)$, an increasing, strictly concave, twice differentiable function of $x_s \geq 0$

Objective: maximize network utility (sum of source utilities)

Economics interpretation: Dual variables as feedback congestion price
NUM Framework

Basic version (A monotropic program):

maximize \( \sum_s U_s(x_s) \)
subject to \( \sum_{s:l \in L(s)} x_s \leq c_l, \ \forall l, \)
\( x \succeq 0 \)

Current applications:

- **Reverse engineering**: TCP congestion control and Internet rate allocation
- **Forward engineering**: Network resource allocation, e.g., power control
- **Layering as optimization decomposition**: TCP/IP/MACPHY interactions

Major approaches:

- **Optimization-theoretic**: distributed optimal solution algorithm
- **Game-theoretic**: Nash equilibrium characterization
Fairness

Family of utility functions parameterized by $\alpha \geq 0$:

$$U^\alpha(x) = \begin{cases} (1 - \alpha)x^{1 - \alpha}, & \text{if } \alpha \neq 1, \\ \log x, & \text{otherwise} \end{cases}$$

$\alpha = 1$: proportional fair

$\alpha = 2$: harmonic-mean fair

$\alpha = \infty$: maxmin fair

Feasible $x$ is proportionally fair (per unit charge) if for any other feasible $x'$,

$$\sum_s w_s \frac{x'_s - x_s}{x_s} \leq 0$$
Lecture 15: TCP Congestion Control Solving NUM

Different source algorithms update primal variables (source rates) for different utilities:

- TCP Vegas: log utilities
- TCP Tahoe: arctan utilities

Different queue management update dual variables (link prices):

- FIFO
- RED

Rigorous and significant implications to equilibrium properties of efficiency, fairness, stability, delay of rate allocation
NUM Extensions

Utility function:
- Nonconcave
- Coupled
- Not a function of rate
- Nonsmooth

Constraint:
- Nonlinear constraints
- Integer constraints

Pricing:
- Per-user differential pricing
**G.NUM**

Generalized NUM:

Minimizing **additive** objective over **additive** constraints

- Compared to NFP: **nonlinear** objectives
- Compared to MP (Basic NUM): **nonlinear** constraints
- Compared to separable convex optimization: **coupling, nonconvexity**
- Compared to general market equilibrium theory: new questions on **distributed solution and dynamic behavior**

A special case of Generalized NUM: **Geometric Programming**
Lecture 16: Layering as Optimization Decomposition

- Network
- Generalized NUM

- Layers
- Decomposed subproblems

- Interfaces
- Functions of primal or dual variables

- Layering
- Decompositions
Lecture 16: Layering as Optimization Decomposition

- How to layer? How not to layer?
- Separation theorem?

- Both reverse engineering and forward engineering
- Systematic study of architectural principles and tradeoffs of layering
- Vertical and horizontal decomposition

How many different ways to decompose?
- Infinite!
- How you write down the problem constraints decomposition possibilities!
Rate allocation constraint set depends on
- Time-varying channel condition
- Adaptive resource allocation, e.g., power control

Joint rate allocation and power control in:
- Cellular: single cell downlink
- Cellular: single cell uplink
- Cellular: multiple cells
- Ad hoc wireless multi-hop networks
- End-to-end hybrid networks

Many other variations: scheduling, beamforming, base station assignment...
Dual Decomposition

Basic NUM:

Convex optimization with zero duality gap

Lagrangian decomposition:

\[ L(x, \lambda) = \sum_s U_s(x_s) + \sum_l \lambda_l \left( c_l - \sum_{s: l \in L(s)} x_s \right) \]

\[ = \sum_s \left[ U_s(x_s) - \left( \sum_{l \in L(s)} \lambda_l \right) x_s \right] + \sum_l c_l \lambda_l \]

\[ = \sum_s L_s(x_s, \lambda^s) + \sum_l c_l \lambda_l \]

Dual problem:

minimize \[ g(\lambda) = L(x^*(\lambda), \lambda) \]

subject to \[ \lambda \succeq 0 \]
Canonical Distributed Algorithm

Source algorithm:

\[ x_s^*(\lambda^s) = \arg\max \left[ U_s(x_s) - \lambda^s x_s \right], \quad \forall s \]

- Selfish net utility maximization locally at source \( s \)

Link algorithm (gradient or subgradient based):

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

- Balancing supply and demand through pricing

Certain choices of step sizes of distributed algorithm guarantee convergence to globally optimal \((x^*, \lambda^*)\)
Kelly’s Decomposition

Problem \( \text{USER}_s(U_s; \lambda_s) \) with variables \( w_s \), one for each source \( s \):

\[
\text{maximize} \quad U_s \left( \frac{w_s}{\lambda_s} \right) - w_s \\
\text{subject to} \quad w_s \geq 0
\]

Source \( s \) chooses to pay \( w_s \) to maximize profit, where \( \lambda_s \) is charge per unit flow for source \( s \)

Problem \( \text{NETWORK}(R, c; w) \) with variables \( x \):

\[
\text{maximize} \quad \sum_s w_s \log x_s \\
\text{subject to} \quad Rx \preceq c \\
\quad \quad \quad x \succeq 0
\]

Network knows payments \( w_s \) from all sources \( r \) and chooses rate allocation to maximize log utility over linear flow constraints

Network does \textbf{not} need to know \( \{U_s\} \), but still needs to be distributively solved
Kelly’s Decomposition

Theorem: there exist \( \lambda, w \) and \( x \) satisfying \( w_s = \lambda_s x_s \) such that

1. \( w_s \) solves \( USER_s(U_s; \lambda_s) \)
2. \( x \) solves \( NETWORK(R, c; w) \)
3. \( x \) is the unique solution to basic NUM

Proof: by Lagrange duality and complementary slackness

Economic interpretation:
- User’s choice of charges and network’s choice of allocated rates and price per unit share reaches equilibrium \( \Rightarrow \) System optimum
- Demand \( w_s \) equals price \( \lambda_s \) times quantity \( x_s \)
Dual NETWORK Problem

Primal problem $NETWORK(R, c; w)$ with variables $x$:

$$\begin{align*}
\text{maximize} & \quad \sum_s w_s \log x_s \\
\text{subject to} & \quad Rx \preceq c \\
& \quad x \succeq 0
\end{align*}$$

Lagrange dual problem $DUAL(R, c; w)$ with variables $\mu$:

$$\begin{align*}
\text{maximize} & \quad \sum_s w_s \log(\sum_{l \in L(s)} \mu_l) - \sum_l \mu_l c_l \\
\text{subject to} & \quad \mu \succeq 0
\end{align*}$$

Recover optimal primal variables from optimal dual variables:

$$x_s = \frac{w_s}{\sum_{l \in L(s)} \mu_l}$$
Primal Algorithm

\[
\frac{dx_s(t)}{dt} = \kappa \left( w_s - x_s \sum_{l \in L(s)} \mu_l(t) \right)
\]

\[
\mu_l(t) = p_l \left( \sum_{s: l \in L(s)} x_s(t) \right)
\]

- Link \( l \) charge \( p_l(y_l) \) per unit flow, when total flow on link \( l \) is \( y_l = \sum_{s: l \in L(s)} x_s \). Each source tries to equalize the total cost with target value \( w_s \).

- Link \( l \) generates feedback signal \( p_l(y_l) \) when total flow on link \( l \) is \( y_l \). Each source linearly increase its rate (proportional to \( w_s \)) and multiplicatively decrease its rate (proportional to total feedback).

Assume \( p_l(y_l) \) is nonnegative, continuous, and increasing function not identically zero.
Global Stability

**Theorem:** The following function is a Lyapunov function for the primal algorithm. The unique $x$ maximizing network utility is a stable point to which all trajectories converge

$$U(x) = \sum_s w_s \log x_s - \sum_l \int_0^{\sum_{s:l \in L(s)} x_s} p_l(y) dy$$

Rates $x$ vary gradually as shadow prices $\mu$ change as functions of $x$

Let

$$p_l(y) = (y - c_l + \epsilon)^+/\epsilon^2$$

As $\epsilon \to 0$, maximization of Lyapunov function approximates arbitrarily closely the primal problem $NETWORK(R, c; w)$

Barrier function interpretation
Lecture Summary

Network Utility Maximization

- An **emerging, unifying** framework for analysis and design of communication systems
- Substantial theoretical advances in recent years
- Significant practical motivations and applications
