Distributed Algorithm Design for Network Optimization Problems with Coupled Objectives: A Tutorial

Dr. Jianwei Huang

Network Communications and Economics Lab (NCEL)

Department of Information Engineering
The Chinese University of Hong Kong

November 2009

When Optimizing Network Performance ...

- Often model as a **NUM** (network utility maximization) problem
  - Utility function: representing performance/satisfaction/happiness
  - Maximizing total utility = maximizing system performance
  - Often want distributed algorithms to achieve global optimal solutions

When Optimizing Network Performance ...

- Often model as a **NUM** (network utility maximization) problem
  - Utility function: representing performance/satisfaction/happiness
  - Maximizing total utility = maximizing system performance
  - Often want distributed algorithms to achieve global optimal solutions
- Solving NUM may not be easy
  - Tight coupling in constraints: primal or dual decompositions
  - Tight coupling in objectives: more difficult to deal with

How to Deal with Coupled Objectives

- Method 1: consistency pricing [TanPalomarChiang'06]
  - Works well with strictly convex problems
  - Require significant message passing
  - Convergence can be slow

How to Deal with Coupled Objectives

- Method 1: consistency pricing [TanPalomarChiang'06]
  - Works well with strictly convex problems
  - Require significant message passing
  - Convergence can be slow
- Method 2: "reverse-engineering" [This Talk]
  - No need for strict convexity
  - Limited message passing
  - Fast and robust convergence

NUM Formulation

- $\mathcal{K} = \{1, \ldots, K\}$: users
- $U_k(x_k, \mathbf{x}_{-k})$: user k's utility
  - x_k: user k's decision variable
  - $\mathbf{x}_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K)$: other users' decision variables
  - Increasing concave in x_k, but not necessarily in $\mathbf{x}_{-k}$
- $\mathcal{X}_k = \left[X_k^{\min}, X_k^{\max}\right]$: feasible set of x_k

NUM Formulation

- $\mathcal{K} = \{1, \ldots, K\}$: users
- $U_k(x_k, \mathbf{x}_{-k})$: user k's utility
  - x_k: user k's decision variable
  - $\mathbf{x}_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K)$: other users' decision variables
  - Increasing concave in x_k, but not necessarily in $\mathbf{x}_{-k}$
- $\mathcal{X}_k = \left[X_k^{\min}, X_k^{\max}\right]$: feasible set of x_k

Definition (NUM)

$$\max_{\mathbf{x} \in \boldsymbol{\mathcal{X}}} \sum_{k \in \mathcal{K}} U_k(x_k, \mathbf{x}_{-k}).$$

NUM Formulation

- $\mathcal{K} = \{1, \ldots, K\}$: users
- $U_k(x_k, \mathbf{x}_{-k})$: user k's utility
  - x_k: user k's decision variable
  - $\mathbf{x}_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K)$: other users' decision variables
  - Increasing concave in x_k, but not necessarily in $\mathbf{x}_{-k}$
- $\mathcal{X}_k = \left[X_k^{\min}, X_k^{\max}\right]$: feasible set of x_k

Definition (NUM)

$$\max_{\mathbf{x} \in \boldsymbol{\mathcal{X}}} \sum_{k \in \mathcal{K}} U_k(x_k, \mathbf{x}_{-k}).$$

- NUM may not be convex optimization problem
  - Difficult to solve even in a centralized fashion
- **Our target**: solve the NUM in a distributed fashion

Starting Point: KKT Condition Set

Definition (KKT Conditions of NUM)

Any global optimal solution $\mathbf{x}^*$ of NUM satisfies the following ($\forall k \in \mathcal{K}$):

$$\begin{aligned} \frac{\partial U_k\left(x_k^*, \mathbf{x}_{-k}^*\right)}{\partial x_k^*} + \sum_{j \neq k} \frac{\partial U_j\left(x_k^*, \mathbf{x}_{-k}^*\right)}{\partial x_k^*} &= \lambda_k^* - \mu_k^*, \\ \lambda_k^*(x_k^* - X_k^{\max}) = 0, \;\; \mu_k^*(X_k^{\min} - x_k^*) &= 0, \\ \lambda_k^*, \mu_k^* &\geq 0. \end{aligned}$$

Starting Point: KKT Condition Set

Definition (KKT Conditions of NUM)

Any global optimal solution $\mathbf{x}^*$ of NUM satisfies the following ($\forall k \in \mathcal{K}$):

$$\begin{aligned}
\frac{\partial U_k\left(x_k^*, \mathbf{x}_{-k}^*\right)}{\partial x_k^*} + \sum_{j \neq k} \frac{\partial U_j\left(x_k^*, \mathbf{x}_{-k}^*\right)}{\partial x_k^*} &= \lambda_k^* - \mu_k^*, \\
\lambda_k^*(x_k^* - X_k^{\max}) = 0, \ \ \mu_k^*(X_k^{\min} - x_k^*) &= 0, \\
\lambda_k^*, \mu_k^* &\geq 0.
\end{aligned}$$

- KKT set contains all KKT points (global & local optimal solutions).
- **Our goal**: design an distributed algorithm to reach the KKT set
  - If KKT set is singleton, then we find the unique global optimal solution.

Key Idea: Localize the Global Problem

- Each user solves a local optimization based on
  - Local observation
  - Limited message passing among users
- Reserve-engineer the proper "local objective function"
  - KKT conditions can be satisfied if all users "properly" optimize their local objectives

Local Objective Function $Y_k(x_k, \mathbf{x}_{-k})$

- Need to satisfy KKT condition:

$$\frac{\partial Y_k(x_k, \mathbf{x}_{-k})}{\partial x_k} = \frac{\partial U_k(x_k, \mathbf{x}_{-k})}{\partial x_k} + \sum_{j \neq k} \frac{\partial U_j(x_k, \mathbf{x}_{-k})}{\partial x_k}.$$

- One (obvious) choice:

$$Y_k(x_k, \mathbf{x}_{-k}) = \sum_{j=1}^{K} U_j(x_k, \mathbf{x}_{-k}).$$

  - May require too much information

Local Objective Function $Z_k(x_k, \mathbf{x}_{-k}, \mathbf{m}_{-k})$

- Construct based on limited message passing
  - $m_k = f_k(\mathbf{x}(t))$: message locally computed and announced by user k
  - $\mathbf{m}_{-k} = (m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_K)$: other users' messages

$$\frac{\partial Z_k(x_k, \mathbf{x}_{-k}, \mathbf{m}_{-k})}{\partial x_k} = \frac{\partial U_k(x_k, \mathbf{x}_{-k})}{\partial x_k} + \sum_{j \neq k} \frac{\partial U_j(x_k, \mathbf{x}_{-k})}{\partial x_k}.$$

- Do not require knowing other users' utility functions

Asynchronous and Distributed Algorithm

Let time $t = 0$.
for each user k
    Randomly initialize $x_k(0)$ and $m_k(0)$.
end
repeat
    $t = t + 1$.
    for each user k
        At a random time, $x_k(t+1) = \arg\max_{x_k} Z_k\left(x_k, \mathbf{x}_{-k}(t), \mathbf{m}_{-k}(t)\right)$.
        At a random time, announce $m_k(t+1) = f_k(\mathbf{x}(t))$.
    end
until $(\mathbf{x}(t), \mathbf{m}(t))$ converge

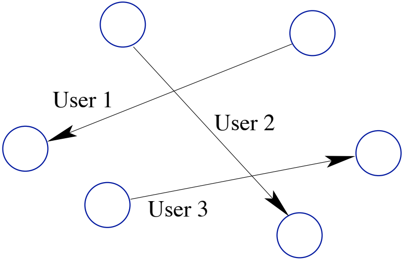
Algorithm Analysis

- Direct method: a mapping based approach
  - Monotone mapping
  - Contraction mapping
- Indirect method: a fictitious game approach
  - Supermodular game
  - Potential game

Case Study I: Optimal Random Access

- A.H. Mohsenian-Rad, J. Huang, M. Chiang and V.W.S. Wong, "Utility-Optimal Random Access: Reduced Complexity, Fast Convergence, and Robust Performance," *IEEE Transactions on Wireless Communications*, Feb. 2009
- A.H. Mohsenian-Rad, J. Huang, M. Chiang and V.W.S. Wong, "Utility-Optimal Random Access: Optimal Performance Without Frequent Explicit Message Passing," *IEEE Transactions on Wireless Communications*, Feb. 2009

(A Simple) Network Model



- A set of $\mathcal{K} = \{1, \ldots, K\}$ single-hop users
- Full interference topology (relaxed later)
- Each user k
  - Contend the channel with probability $p_k \in \mathcal{P}_k = [P_k^{\min}, P_k^{\max}]$
  - Maximum data rate γ_k
  - Long term average data rate

$$r_k(\mathbf{p}) = \gamma_k p_k \prod_{j \in \mathcal{K} \setminus \{k\}} (1 - p_j)$$

Network Utility Maximization

- Each user k has an increasing and concave utility function $U_k(r_k)$

Definition (NUM)

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{k\in\mathcal{N}} U_k(r_k(\mathbf{p})),$$

Network Utility Maximization

- Each user k has an increasing and concave utility function $U_k(r_k)$

Definition (NUM)

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{k\in\mathcal{N}} U_k(r_k(\mathbf{p})),$$

- We will focus on the α-fair utility function:

$$U_k(x) = \begin{cases} (1-\alpha)^{-1}\ x^{1-\alpha}, & \text{if } \alpha \in (0,1)\bigcup(1,\infty), \\ \log x, & \text{if } \alpha = 1. \end{cases}$$

  - $\alpha \to 0$: system throughput maximization
  - $\alpha = 1$: proportional fair allocation
  - $\alpha \to \infty$: max-min fairness

Local Optimization Problem

Definition (Local NUM)

$$\max_{p_k \in \mathcal{P}_k} \sum_{j \in \mathcal{K}} U_j(r_j(p_k, \mathbf{p}_{-k})),$$

- Simple to solve.

Optimal Solution of Local Optimization

$$p_k^*(\mathbf{p}_{-k}) = f_k(\mathbf{p}_{-k}) = \left[1/\left(1 + \sqrt[\alpha]{v_k(\mathbf{p}_{-k})}\right)\right]_{P_k^{\min}}^{P_k^{\max}},$$

Optimal Solution of Local Optimization

$$p_k^*(\mathbf{p}_{-k}) = f_k(\mathbf{p}_{-k}) = \left[1/\left(1 + \sqrt[\alpha]{v_k(\mathbf{p}_{-k})}\right)\right]_{P_k^{\min}}^{P_k^{\max}},$$

- $v_k(\mathbf{p}_{-k}) = {\gamma_k}^{\alpha-1} \sum_{j \in \mathcal{K} \setminus \{k\}} m_j$

Optimal Solution of Local Optimization

$$p_k^*(\mathbf{p}_{-k}) = f_k(\mathbf{p}_{-k}) = \left[1/\left(1 + \sqrt[\alpha]{v_k(\mathbf{p}_{-k})}\right)\right]_{P_k^{\min}}^{P_k^{\max}},$$

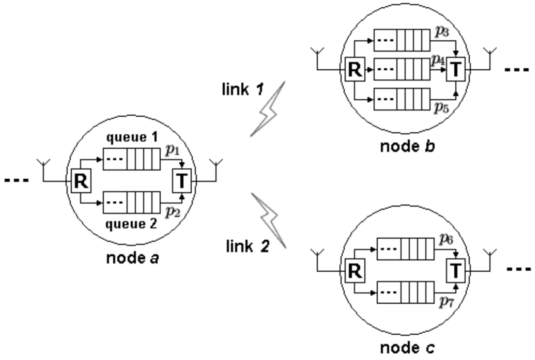
- $v_k(\mathbf{p}_{-k}) = {\gamma_k}^{\alpha-1} \sum_{j\in\mathcal{K}\setminus\{k\}} m_j$
- $m_j = (1/\gamma_j)^{\alpha-1} (1/p_j - 1)^{\alpha-1}, \quad \forall j \in \mathcal{K}$

Optimal Solution of Local Optimization

$$p_k^*(\mathbf{p}_{-k}) = f_k(\mathbf{p}_{-k}) = \left[1 / \left(1 + \sqrt[\alpha]{v_k(\mathbf{p}_{-k})} \right) \right]_{P_k^{\min}}^{P_k^{\max}},$$

- $v_k(\mathbf{p}_{-k}) = {\gamma_k}^{\alpha-1} \sum_{j \in \mathcal{K} \setminus \{k\}} m_j$
- $m_j = (1/\gamma_j)^{\alpha-1} (1/p_j - 1)^{\alpha-1}, \quad \forall j \in \mathcal{K}$
- We have a customized algorithm by plugging in p_k and m_j updates.

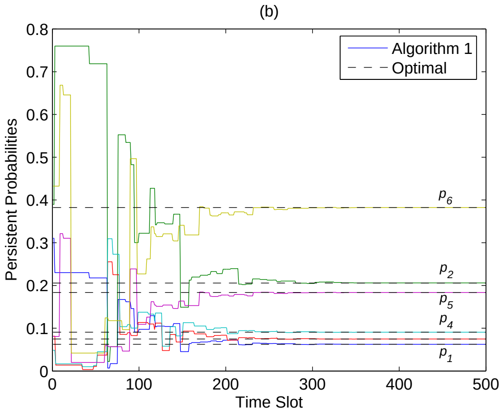
Extensions



- Can also be extended to general interference case
  - A node can have multiple outgoing links
  - A link may only interfere with a subset of other links

Numerical Example

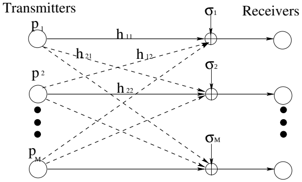
- 3 nodes and 6 links; $\alpha = 0.6$ (non-convex NUM)



Case Study II: Optimal Power Control

- J. Huang, R. Berry and M. L. Honig, "Distributed Interference Compensation for Wireless Networks," *IEEE Journal on Selected Areas in Communications*, May 2006

Network Model



- A set of $\mathcal{K} = \{1, ..., K\}$ users.
- For each user $k \in \mathcal{K}$:
  - Transmit power constraint: $p_k \in [P_k^{min}, P_k^{max}]$.
  - Received **SINR** (signal-to-interference plus noise ratio):

$$\gamma_k(\mathbf{p}) = \frac{h_{k,k} p_k}{\sigma_k + \sum_{j \neq k} h_{k,j} p_j}.$$

  - **Utility function** $U_k(\gamma_k)$: increasing, differentiable, strictly concave.

NUM Problem

Definition (NUM)

$$\max_{\mathbf{p}\in\mathcal{P}} \sum_{k} U_k(\gamma_k(\mathbf{p})).$$

Algorithm

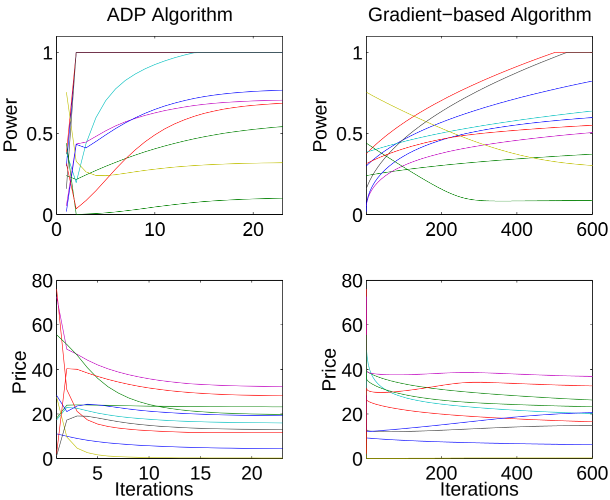
- **Message (Price) Announcing**: user k announces price:

$$\pi_k = \left| \frac{\partial U_k(\gamma_k)}{\partial I_k} \right| = \frac{\partial U_k(\gamma_n)}{\partial \gamma_k} \frac{\gamma_k^2}{p_k h_{k,k}}.$$

- **Power Updating**: user k updates power p_k to maximize surplus:

$$Y_k(p_k, \mathbf{p}_k, \boldsymbol{\pi}_{-k}) = U_k(\gamma_k(p_k, \mathbf{p}_{-k})) - p_k \sum_{j \neq k} \pi_j h_{j,k}.$$

Numerical Example



log utilities, 10 users, Gradient-based Algorithm is based on [Chiang'05]

Conclusions

- **Framework**: design distributed algorithm for NUM with coupled objectives
- **Ke Ideas**:
  - Reverse-engineering
  - Step-size free updates
  - Limited message passing
- **Examples**:
  - Optimal random access
  - Optimal power control

Contact

Network Communications and Economics Lab (NCEL)

www.ie.cuhk.edu.hk/$\sim$jwhuang

jwhuang@ie.cuhk.edu.hk