Optimal Resource Allocation in a Multi-User Interference Channel: Complexity Analysis and Algorithm Design

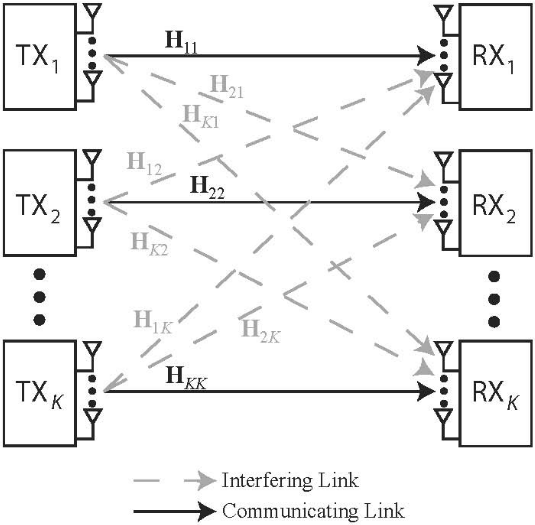
Ya-Feng Liu

State Key Laboratory of Scientific/Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, China

October 22, 2014, INC/IE, CUHK, Hong Kong

OPTIMAL RESOURCE ALLOCATION PROBLEM

Multi-User Interference Channel (IC)



System Model

- K users or links (transmitter-receiver pairs) and denote $\mathcal{K} = \{1, 2, \ldots, K\}$
- Assume each transmitter sends one data stream s_k to its intended receiver
- M_k and N_j denote the number of antennas at receiver k and transmitter j
- $\mathbf{H}_{kj} \in \mathbb{C}^{M_k \times N_j}$ is the channel matrix from transmitter j to receiver k
  - $M_k = 1, N_k = 1 \longrightarrow$ SISO IC
  - $M_k = 1, N_k \geq 2 \longrightarrow$ MISO IC
  - $M_k \geq 2, N_k = 1 \longrightarrow$ SIMO IC
  - $M_k \geq 2, N_k \geq 2 \longrightarrow$ MIMO IC

SINR and Rate

- Linear transmission and reception strategy
- $\mathbf{u}_k$ and $\mathbf{v}_j$ are the beamforming vectors at receiver k and transmitter j
- User k's received signal:

$$\hat{s}_k = \underbrace{\mathbf{u}_k^\dagger \mathbf{H}_{kk} \mathbf{v}_k s_k}_{\text{signal term}} + \underbrace{\sum_{j \neq k} \mathbf{u}_k^\dagger \mathbf{H}_{kj} \mathbf{v}_j s_j}_{\text{interference term}} + \underbrace{\mathbf{u}_k^\dagger \mathbf{n}_k}_{\text{noise term}}$$

- SINR at receiver k (MIMO/SISO):

$$\text{SINR}_k = \frac{|\mathbf{u}_k^\dagger \mathbf{H}_{kk} \mathbf{v}_k|^2}{\sum_{j \neq k} |\mathbf{u}_k^\dagger \mathbf{H}_{kj} \mathbf{v}_j|^2 + \eta_k \|\mathbf{u}_k\|^2}, \quad \text{SINR}_k = \frac{g_{kk} p_k}{\sum_{j \neq k} g_{kj} p_j + \eta_k}$$

- Transmission rate: $r_k = \log_2 (1 + \text{SINR}_k)$

Optimal Resource Allocation Problems

Optimal resource allocation problems in the multi-user interference channel are to design $\{\mathbf{u}_k, \mathbf{v}_k\}$ to achieve some goals under some constraints. They are often formulated as system utility maximization problems subject to power budget constraints, or total power minimization problems subject to QoS constraints.

Two Questions

Given an optimal resource allocation problem,

Q1: Is there any polynomial time algorithm which can solve it to global optimality?

Q2: How to design a "good" algorithm for solving it with guaranteed performance?

Two Questions

Given an optimal resource allocation problem,

Q1: Is there any polynomial time algorithm which can solve it to global optimality?

Q2: How to design a "good" algorithm for solving it with guaranteed performance?

A1: Study the computational complexity of the problem and identify polynomial time solvable subclass (if the general problem is "hard")!

A2: Design customized algorithms for the problem by fully taking advantage of its special structures such as (hidden) convexity, separability, and nonnegativity!

Complexity Theory

- Convexity versus nonconvexity

Complexity Theory

- Convexity versus nonconvexity
- Complexity theory: a robust tool to characterize the computational tractability of an optimization problem.
- Once a problem is shown to be “hard”, the search for an efficient, exact algorithm should be accorded low priority or avoided.

Complexity Theory

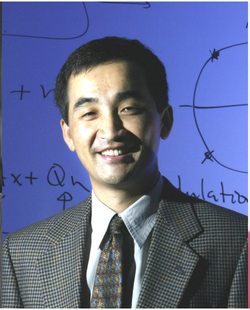
- Convexity versus nonconvexity
- Complexity theory: a robust tool to characterize the computational tractability of an optimization problem.
- Once a problem is shown to be "hard", the search for an efficient, exact algorithm should be accorded low priority or avoided.
- Concentrate on other less ambitious approaches
  - look for efficient algorithms that solve various special cases of the general problem
  - look for algorithms that, though not guaranteed to run quickly, seem likely to do so most of the time
  - relax the problem somewhat, looking for a fast algorithm that finds a solution merely satisfying most of constraints [Luo-Ma-So-Ye-Zhang, SPM, 2010]

Outline

- Part I: Linear Transceiver Design for a MIMO IC
  - Complexity Analysis
  - Algorithm Design
  - Simulation Results
- Part II: Joint Power and Admission Control for a SISO IC
  - Complexity Analysis
  - Algorithm Design
  - Simulation Results
- Concluding Remarks

My Collaborators

Yu-Hong Dai



Tom Luo

Mingyi Hong

Shiqian Ma

PART I: LINEAR TRANSCEIVER DESIGN

Max-Min Fairness Linear Transceiver Design

$$\max_{\{\mathbf{u}_k,\mathbf{v}_k\}} \quad \min_{k\in\mathcal{K}} \left\{ \mathsf{SINR}_k := \frac{|\mathbf{u}_k^\dagger \mathbf{H}_{kk}\mathbf{v}_k|^2}{\sum_{j\neq k} |\mathbf{u}_k^\dagger \mathbf{H}_{kj}\mathbf{v}_j|^2 + \eta_k \|\mathbf{u}_k\|^2} \right\}$$

$$\text{s.t.} \quad \|\mathbf{u}_k\| = 1,\ \|\mathbf{v}_k\|^2 \leq \bar{p}_k,\ k \in \mathcal{K}$$

$$\Updownarrow$$

$$\max_{\{\mathbf{u}_k,\mathbf{v}_k,\zeta\}} \quad \zeta$$

$$\text{s.t.} \quad \mathsf{SINR}_k \geq \zeta, \|\mathbf{u}_k\| = 1, \|\mathbf{v}_k\|^2 \leq \bar{p}_k,\ k \in \mathcal{K}$$

Strongly NP-Hard: MIMO Case

Theorem (L.-Dai-Luo, ICC, 2011)

Given a SINR target ζ, checking the feasibility of problem

$$\begin{cases} SINR_k \geq \zeta, \; k \in \mathcal{K} \\ \|\mathbf{u}_k\| = 1, \; k \in \mathcal{K} \\ \|\mathbf{v}_k\|^2 \leq P_k, \; k \in \mathcal{K} \end{cases}$$

is strongly NP-hard for the MIMO interference channel with

$$M_k \geq 3, N_k \geq 2, \;\forall\; k \in \mathcal{K}$$

or

$$M_k \geq 2, N_k \geq 3, \;\forall\; k \in \mathcal{K}.$$

- Feasibility $\longrightarrow$ Optimization

Polynomial Time Solvable: MISO Case

Theorem (Wiesel-Eldar-Shitz, TSP, 2006; L.-Dai-Luo, TSP, 2011)

The max-min fairness linear transceiver design problem for the MISO interference channel

$$\begin{aligned} \max_{\{\mathbf{v}_k\}} \quad & \min_{k\in\mathcal{K}} \left\{ \frac{|\mathbf{h}_{kk}^{\dagger}\mathbf{v}_k|^2}{\sum_{j\neq k} |\mathbf{h}_{kj}^{\dagger}\mathbf{v}_j|^2 + \eta_k} \right\} \\ s.t. \quad & \|\mathbf{v}_k\|^2 \leq \bar{p}_k, \ \ k \in \mathcal{K} \end{aligned}$$

is polynomial time solvable.

Polynomial Time Solvable: MISO Case

Theorem (Wiesel-Eldar-Shitz, TSP, 2006; L.-Dai-Luo, TSP, 2011)

The max-min fairness linear transceiver design problem for the MISO interference channel

$$\begin{aligned} \max_{\{\mathbf{v}_k\}} \quad & \min_{k\in\mathcal{K}} \left\{ \frac{|\mathbf{h}_{kk}^\dagger \mathbf{v}_k|^2}{\sum_{j\neq k} |\mathbf{h}_{kj}^\dagger \mathbf{v}_j|^2 + \eta_k} \right\} \\ s.t. \quad & \|\mathbf{v}_k\|^2 \le \bar{p}_k,\ \ k\in\mathcal{K} \end{aligned}$$

is polynomial time solvable.

- Key observation:

$$\mathbf{h}_{kk}^\dagger \mathbf{v}_k \ge \zeta \sqrt{\sum_{j\neq k} |\mathbf{h}_{kj}^\dagger \mathbf{v}_j|^2 + \eta_k},\ \ k\in\mathcal{K}$$

Polynomial Time Solvable: SIMO Case

Theorem (L.-Hong-Dai, SPL, 2013)

The max-min fairness linear transceiver design problem for the SIMO interference channel

$$\begin{aligned} \max_{\{\mathbf{u}_k, p_k\}} \quad & \min_{k \in \mathcal{K}} \left\{ \frac{|\mathbf{u}_k^\dagger \mathbf{h}_{kk}|^2 p_k}{\sum_{j \neq k} |\mathbf{u}_k^\dagger \mathbf{h}_{kj}|^2 p_j + \eta_k \|\mathbf{u}_k\|^2} \right\} \\ s.t. \quad & 0 \leq p_k \leq \bar{p}_k,\ k \in \mathcal{K} \end{aligned}$$

is polynomial time solvable.

Polynomial Time Solvable: SIMO Case

Theorem (L.-Hong-Dai, SPL, 2013)

The max-min fairness linear transceiver design problem for the SIMO interference channel

$$\begin{aligned} \max_{\{\mathbf{u}_k, p_k\}} \quad & \min_{k\in\mathcal{K}} \left\{ \frac{|\mathbf{u}_k^\dagger \mathbf{h}_{kk}|^2 p_k}{\sum_{j\neq k} |\mathbf{u}_k^\dagger \mathbf{h}_{kj}|^2 p_j + \eta_k \|\mathbf{u}_k\|^2} \right\} \\ \text{s.t.} \quad & 0 \le p_k \le \bar{p}_k,\ k \in \mathcal{K} \end{aligned}$$

is polynomial time solvable.

- There is hidden convexity in the problem!
- SDPBA globally solves the above problem in polynomial time!

Complexity Status

Complexity Status of Max-Min Fairness Linear Transceiver Design

Rx \ Tx	$N_k = 1$	$N_k = 2$	$N_k \geq 3$
$M_k = 1$	Poly. Time Alg.	Poly. Time Alg.	Poly. Time Alg.
$M_k = 2$	Poly. Time Alg.	MISSING CASE	Str. NP-hard
$M_k \geq 3$	Poly. Time Alg.	Str. NP-hard	Str. NP-hard

Complexity Status

Complexity Status of Max-Min Fairness Linear Transceiver Design

Rx \ Tx	$N_k = 1$	$N_k = 2$	$N_k \geq 3$
$M_k = 1$	Poly. Time Alg.	Poly. Time Alg.	Poly. Time Alg.
$M_k = 2$	Poly. Time Alg.	MISSING CASE	Str. NP-hard
$M_k \geq 3$	Poly. Time Alg.	Str. NP-hard	Str. NP-hard

- The complexity is sensitive to the number of antennas!

Complexity Status

Complexity Status of Max-Min Fairness Linear Transceiver Design

Rx \ Tx	$N_k = 1$	$N_k = 2$	$N_k \geq 3$
$M_k = 1$	Poly. Time Alg.	Poly. Time Alg.	Poly. Time Alg.
$M_k = 2$	Poly. Time Alg.	MISSING CASE	Str. NP-hard
$M_k \geq 3$	Poly. Time Alg.	Str. NP-hard	Str. NP-hard

- The complexity is sensitive to the number of antennas!
- The problem remains NP-hard in this case [Razaviyayn-Hong-Luo, SP, 2013].

A Framework of Cyclic Coordinate Ascent Algorithm

for the Linear Transceiver Design Problem

Step1. Given $\mathbf{v}^0$ and set $n = 0$.

Step2. Fixing $\{\mathbf{v}_k^n\}_{k\in\mathcal{K}}$, compute the receive beamformers $\{\mathbf{u}_k^n\}_{k\in\mathcal{K}}$.

Step3. Fixing $\{\mathbf{u}_k^n\}_{k\in\mathcal{K}}$, compute the transmit beamformers $\left\{\mathbf{v}_k^{n+1}\right\}_{k\in\mathcal{K}}$.

Step4. If stopping criterion is satisfied, terminate the algorithm; else set $n = n + 1$ and go to **Step2**.

- $n \geq 0$ denotes the iteration index.
- Partition the variables into two blocks $\mathbf{u} = (\mathbf{u}_1; ...; \mathbf{u}_K)$ and $\mathbf{v} = (\mathbf{v}_1; ...; \mathbf{v}_K)$ to exploit the separable structures.

Step2: Fixing $\mathbf{v} = \mathbf{v}^n$

- Fixing $\mathbf{v} = \mathbf{v}^n$, the problem with respect to $\mathbf{u}$ can be solved by solving K independent small problems

$$\begin{aligned} \max_{\{\mathbf{u}_k\}} \quad & \frac{|\mathbf{u}_k^\dagger \mathbf{H}_{kk} \mathbf{v}_k^n|^2}{\sum_{j \neq k} |\mathbf{u}_k^\dagger \mathbf{H}_{kj} \mathbf{v}_j^n|^2 + \eta_k \|\mathbf{u}_k\|^2} \\ \text{s.t.} \quad & \|\mathbf{u}_k\|^2 = 1 \end{aligned}$$

- The LMMSE receive beamformers

$$\mathbf{u}_k^n = \tilde{\mathbf{u}}_k^n / \|\tilde{\mathbf{u}}_k^n\|, \ \tilde{\mathbf{u}}_k^n = \left(\sum_{j=1}^{K} \mathbf{H}_{kj} \mathbf{v}_j^n \left(\mathbf{H}_{kj} \mathbf{v}_j^n \right)^\dagger + \eta_k \mathbf{I} \right)^{-1} \mathbf{H}_{kk} \mathbf{v}_k^n, \ k \in \mathcal{K}$$

STEP3: FIXING $\mathbf{u} = \mathbf{u}^n$

- Fixing $\{\mathbf{u}_k^n\}_{k\in\mathcal{K}}$, solve the optimal transmit beamformers $\{\mathbf{v}_k^{n+1}\}_{k\in\mathcal{K}}$ via

$$\max_{\{\mathbf{v}_k\}} \quad \min_{k\in\mathcal{K}} \left\{ \frac{|\left(\mathbf{u}_k^n\right)^\dagger \mathbf{H}_{kk}\mathbf{v}_k|^2}{\sum_{j\neq k} |\left(\mathbf{u}_k^n\right)^\dagger \mathbf{H}_{kj}\mathbf{v}_j|^2 + \eta_k} \right\}$$

$$\text{s.t.} \quad \|\mathbf{v}_k\|^2 \leq \bar{p}_k, \; k \in \mathcal{K}$$

- The above problem can be solved to global optimality in polynomial time using a bisection procedure, where each step solves a second order cone programming (SOCP) [L.-Dai-Luo, TSP, 2011].

- Therefore, the exact CCAA (ECCAA) decomposes the original problem into a series of "simple" subproblems, which can be solved efficiently.

Convergence of ECCAA

- CCA (or BCD) has been widely (or wildly?) used.
- In general, cyclic coordinate algorithms may not converge to a KKT solution even if each subproblem is exactly solved [Powell, MP, 1976].
- A convergence result [Bertsekas, 1999] for this algorithm requires:
  - the constraints are separable ($\surd$)
  - the objective function is continuously differentiable ($\times$)
  - each subproblem has a unique solution ($\times$)

Convergence of ECCAA

- CCA (or BCD) has been widely (or wildly?) used.
- In general, cyclic coordinate algorithms may not converge to a KKT solution even if each subproblem is exactly solved [Powell, MP, 1976].
- A convergence result [Bertsekas, 1999] for this algorithm requires:
  - the constraints are separable ($\surd$)
  - the objective function is continuously differentiable ($\times$)
  - each subproblem has a unique solution ($\times$)
- Global convergence of ECCAA [L.-Dai-Luo, ICC, 2011]

Theorem

The sequence $\{(\mathbf{u}^n, \mathbf{v}^n)\}$ generated by the ECCAA either terminates at a stationary point or any of its accumulation point is a stationary point of the original problem.

A Low-Complexity Algorithm: ICCAA

- Motivation for inexact CCAA (ICCAA)
  - In ECCAA, updating $\mathbf{v}^{n+1}$ requires solving a sequence of SOCP feasibility problems and hence is computationally expensive.
  - Design a scheme for updating $\mathbf{v}^{n+1}$ with a moderate computation cost.
- In ICCAA, we propose to update $\mathbf{v}^{n+1}$ by solving

$$\begin{aligned} \max_{\{\mathbf{v},\theta\}} \quad & \theta \\ \text{s.t.} \quad & \frac{(\mathbf{u}_k^n)^\dagger \mathbf{H}_{kk}\mathbf{v}_k - \theta}{\sqrt{\eta_k + \sum_{j\neq k} |(\mathbf{u}_k^n)^\dagger \mathbf{H}_{kj}\mathbf{v}_j|^2}} \geq \sqrt{G_{2n}},\ k \in \mathcal{K}, \\ & \|\mathbf{v}_k\|^2 \leq \bar{p}_k,\ k \in \mathcal{K}, \end{aligned} \tag{1}$$

where G_{2n} is the minimum SINR value among all users at point $(\mathbf{u}^n, \mathbf{v}^n)$.

ICCAA

- ICCAA: the solution to problem (1) does not solve the subproblem in ECCAA in general.
- Problem (1) is an SOCP and hence can be solved to global optimality in polynomial time.
- ECCAA versus ICCAA

ICCAA

- ICCAA: the solution to problem (1) does not solve the subproblem in ECCAA in general.
- Problem (1) is an SOCP and hence can be solved to global optimality in polynomial time.
- ECCAA versus ICCAA
- Global convergence of ICCAA [L.-Dai-Luo, TSP, 2013]

Theorem

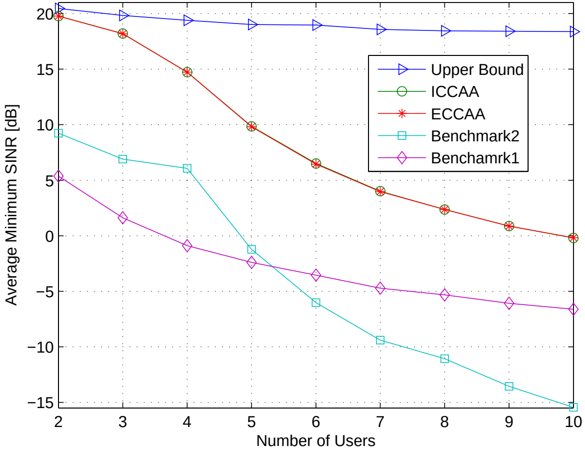
The sequence $\{(\mathbf{u}^n, \mathbf{v}^n)\}$ generated by the ICCAA either terminates at a stationary point or any of its accumulation point is a stationary point of the original problem.

Simulation Setup

- Two scenarios
  - MIMO interference channel with $M_k = 2$ and $N_k = 3$ for all $k \in \mathcal{K}$
  - SIMO interference channel with $M_k = 4$ and $N_k = 1$ for all $k \in \mathcal{K}$
- Channel matrix (vector) $\mathbf{H}_{kj} \sim \mathcal{CN}(0, 1)$
- SNR$= -10 \log_{10}(\eta)$
- 200 independent channel realizations
- Comparison criteria
  - minimum SINR
  - CPU time

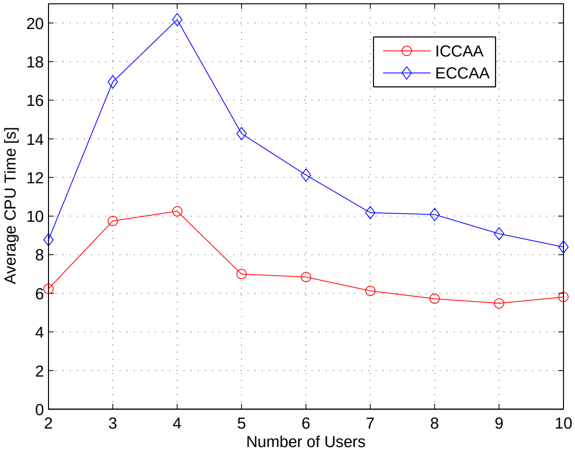
SINR VS NUMBER OF USERS (MIMO SCENARIO)

Average minimum SINR versus the number of users with SNR = 15 dB



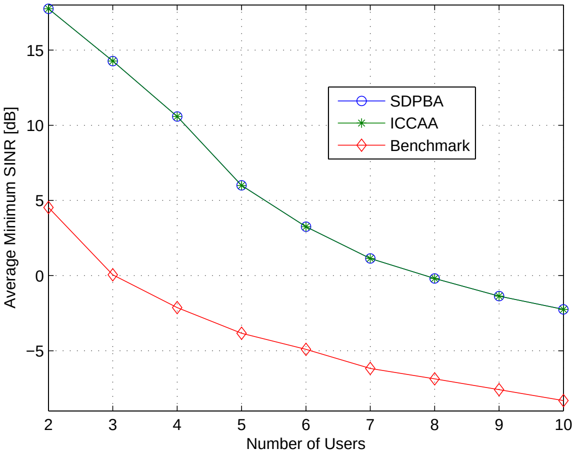
CPU time VS number of users (MIMO Scenario)

Average CPU time versus the number of users with SNR = 15 dB



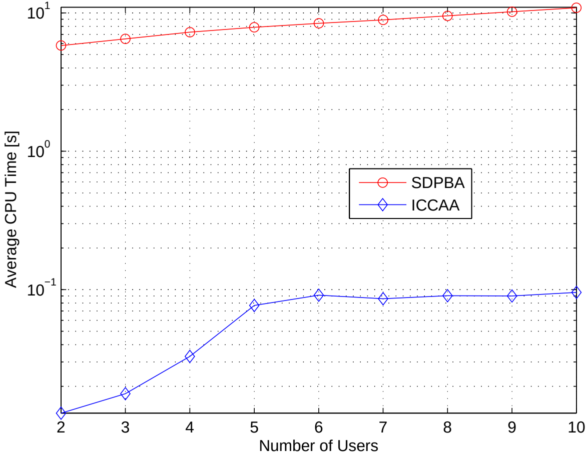
SINR VS NUMBER OF USERS (SIMO SCENARIO)

Average minimum SINR versus the number of users with SNR = 15 dB



CPU time VS number of users (SIMO Scenario)

Average CPU time versus the number of users with SNR = 15 dB



PART II: JOINT POWER AND ADMISSION CONTROL

Power Control

- Power control problem

$$\begin{aligned} \min \quad & \mathbf{e}^T \mathbf{p} \\ \text{s.t.} \quad & \frac{g_{kk} p_k}{\sum_{j \neq k} g_{kj} p_j + \eta_k} \geq \gamma_k, \ \ k \in \mathcal{K} \\ & \mathbf{0} \leq \mathbf{p} \leq \bar{\mathbf{p}} \end{aligned}$$

  - $\mathbf{e} = (1, 1, \ldots, 1)^T$
  - $\mathbf{p} = (p_1, p_2, \cdots, p_K)^T$
  - $\bar{\mathbf{p}} = (\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_K)^T$

- The Foschini-Miljanic algorithm [Foschini-Miljanic, TVT, 1993]

Joint Power and Admission Control (JPAC)

- INFEASIBILITY ISSUE of the linear system

$$\mathrm{SINR}_k \geq \gamma_k, \ \bar{p}_k \geq p_k \geq 0, \ k \in \mathcal{K}$$

- The admission control is necessary to determine the connections to be removed.

Joint Power and Admission Control (JPAC)

- INFEASIBILITY ISSUE of the linear system

$$\text{SINR}_k \geq \gamma_k, \ \ \bar{p}_k \geq p_k \geq 0, \ \ k \in \mathcal{K}$$

- The admission control is necessary to determine the connections to be removed.
- Joint power and admission control (JPAC)
  - the admitted links should be satisfied with their required SINR targets
  - the number of admitted (removed) links should be maximized (minimized)
  - the total transmission power to support the admitted links should be minimized

Two-Stage Formulation

- A two-stage optimization problem:
  - maximize the number of admitted links (with prescribed SINR targets):

$$\begin{aligned} \max_{\mathbf{p},\,\mathcal{S}} \quad & |\mathcal{S}| \\ \text{s.t.} \quad & \mathrm{SINR}_k \geq \gamma_k, \ \ k \in \mathcal{S} \subseteq \mathcal{K} \\ & \mathbf{0} \leq \mathbf{p} \leq \bar{\mathbf{p}} \end{aligned} \tag{2}$$

  - use $\mathcal{S}^*$ to denote the maximum admissible set for problem (2), and $\mathcal{S}^*$ might NOT be unique
  - minimize the total transmission power required to support the admitted links:

$$\begin{aligned} \min \quad & \textstyle\sum_{k \in \mathcal{S}^*} p_k \\ \text{s.t.} \quad & \mathrm{SINR}_k \geq \gamma_k, \ \ k \in \mathcal{S}^* \\ & 0 \leq p_k \leq \bar{p}_k, \ \ k \in \mathcal{S}^* \end{aligned} \tag{3}$$

Complexity Results

- Power control problem (3) is feasible, and can be efficiently solved [Foschini-Miljanic, TVT, 1993]

- However, admission control problem (2) of finding the maximum admissible set $\mathcal{S}^*$ is NP-hard [Mitliagkas-Sidiropoulos-Swami, TWC, 2011]

- The complexity result guides us to develop heuristic algorithms for the JPAC problem.

Existing Algorithms

- Removal-based algorithms
  - update the power, and check whether all links in the network can be supported
  - if yes, terminate the algorithm
  - if not, remove one link from the network, and update the power again

Existing Algorithms

- Removal-based algorithms
  - update the power, and check whether all links in the network can be supported
  - if yes, terminate the algorithm
  - if not, remove one link from the network, and update the power again
- Three key steps
  - power update
  - feasibility check
  - link removal

Normalized Channel

- Two equivalent equations:
  - power constraint: $0 \le p_k \le \bar{p}_k \Leftrightarrow 0 \le x_k := \frac{p_k}{\bar{p}_k} \le 1$
  - SINR constraint: $\frac{g_{kk} p_k}{\sum_{j \ne k} g_{kj} p_j + \eta_k} \ge \gamma_k \Leftrightarrow \frac{1 x_k}{\sum_{j \ne k} \frac{\gamma_k g_{kj} \bar{p}_j}{g_{kk} \bar{p}_k} x_j + \frac{\gamma_k \eta_k}{g_{kk} \bar{p}_k}} \ge 1$
- Normalized Channel:
  - noise vector $\mathbf{b} = \left(\frac{\gamma_1 \eta_1}{g_{11} \bar{p}_1}, \frac{\gamma_2 \eta_2}{g_{22} \bar{p}_2}, \ldots, \frac{\gamma_K \eta_K}{g_{KK} \bar{p}_K} \right)^T > \mathbf{0}$
  - power allocation vector $\mathbf{x} = \left(\frac{p_1}{\bar{p}_1}, \frac{p_2}{\bar{p}_2}, \ldots, \frac{p_K}{\bar{p}_K} \right)$
  - channel gain matrix $\mathbf{A}$ with its (k, j)-th entry

$$a_{kj} = \begin{cases} -\frac{\gamma_k g_{kj} \bar{p}_j}{g_{kk} \bar{p}_k}, & \text{if } k \ne j; \\ 1, & \text{if } k = j. \end{cases}$$

Normalized Channel

With this normalization:

- Focus on **A** and **b**
- $\mathrm{SINR}_k \geq \gamma_k \Leftrightarrow [\mathbf{Ax} - \mathbf{b}]_k \geq 0$

Normalized Channel

With this normalization:

- Focus on $\mathbf{A}$ and $\mathbf{b}$
- $\text{SINR}_k \geq \gamma_k \Leftrightarrow [\mathbf{Ax} - \mathbf{b}]_k \geq 0$
- Special Structure of $\mathbf{A}$ and $\mathbf{b}$ $\Rightarrow$ Algorithm Design

L_0-Minimization Reformulation

Theorem (L.-Dai-Luo, TSP, 2013)

The two-stage JPAC problem can be equivalently reformulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_0 + \alpha\,\bar{\mathbf{p}}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{aligned} \tag{4}$$

where

$$0 < \alpha < \alpha_1 := 1/\bar{\mathbf{p}}^T\mathbf{e}.$$

- Problem (4) can find the maximum admissible set $\mathcal{S}^*$ and at the same time minimize the total required transmission power to support the links in $\mathcal{S}^*$.

L_0-Minimization Reformulation

Theorem (L.-Dai-Luo, TSP, 2013)

The two-stage JPAC problem can be equivalently reformulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_0 + \alpha\, \bar{\mathbf{p}}^T \mathbf{x} \\ s.t. \quad & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{aligned} \tag{4}$$

where

$$0 < \alpha < \alpha_1 := 1/\bar{\mathbf{p}}^T \mathbf{e}.$$

- Problem (4) can find the maximum admissible set $\mathcal{S}^*$ and at the same time minimize the total required transmission power to support the links in $\mathcal{S}^*$.
- Problem (4) is capable of picking the maximum admissible set with minimum total transmission power among potentially many maximum admissible sets.
- Better formulation

Linear Programming Approximation

- L_1-convex approximation

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_1 + \alpha\,\bar{\mathbf{p}}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{aligned}$$

- We further show L_1-minimization problem is equivalent to the following linear program (LP)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{e}^T(\mathbf{b}-\mathbf{A}\mathbf{x}) + \alpha\,\bar{\mathbf{p}}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{b}-\mathbf{A}\mathbf{x} \ge \mathbf{0} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{e} \end{aligned} \tag{5}$$

  - the quantity, $\mathbf{x}_k^e = [\mathbf{b}-\mathbf{A}\mathbf{x}]_k$, measures the excess transmission power
  - LP (5) actually minimizes a weighted sum of the total excess transmission power and the total real transmission power

A Stone and Three Birds I

- Power control
  - power allocation is given by

$$p_k = \bar{p}_k x_k, \ \ k \in \mathcal{K}$$

- Feasibility check
  - recall $\text{SINR}_k \geq \gamma_k \Leftrightarrow [\mathbf{A}\mathbf{x} - \mathbf{b}]_k \geq 0$
  - solve LP (5) with an appropriate $\alpha > 0$ and check $\mathbf{A}\mathbf{x} - \mathbf{b}$
  - $\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$ if and only if all links can be simultaneously supported

A Stone and Three Birds II

- Link removal
  - having obtained the solution of LP (5), we can use the same idea in [Mitliagkas-Sidiropoulos-Swami, TWC, 2011], i.e., drop link k_0 with

$$k_0 = \arg\max_{k \in \mathcal{K}} \left\{ \sum_{j \neq k} |a_{jk}| x_k^e + \sum_{j \neq k} |a_{kj}| x_j^e \right\} \tag{6}$$

  - the above removal strategy can be rewritten as

$$\sum_{j \neq k} |a_{jk}| x_k^e + \sum_{j \neq k} |a_{kj}| x_j^e = \sum_{j \neq k} \frac{\gamma_j}{g_{jj} \bar{p}_j} g_{jk} p_k^e + \sum_{j \neq k} \frac{\gamma_k}{g_{kk} \bar{p}_k} g_{kj} p_j^e$$

  - different from the removal scheme in [Mitliagkas-Sidiropoulos-Swami, TWC, 2011]

$$k_0 = \arg\max_{k \in \mathcal{K}} \left\{ \sum_{j \neq k} g_{jk} p_k^e + \sum_{j \neq k} g_{kj} p_j^e \right\} \tag{7}$$

NLPD Algorithm [L.-Dai-Luo, TSP, 2013]

A New Linear Programming Deflation Algorithm

Step 1. Initialization: Input data $(\mathbf{A}, \mathbf{b}, \bar{\mathbf{p}})$ and set $\mathcal{S} = \mathcal{K}$.

Step 2. Preprocessing (not covered).

Step 3. Power control: Solve linear program (5); check whether all links are supported: if yes, go to **Step 5**; else go to **Step 4**.

Step 4. Admission control: Remove link k_0 according to (6), set $\mathcal{S} = \mathcal{S} \setminus \{k_0\}$, and go to **Step 3**.

Step 5. Postprocessing: Check the removed links for possible admission.

NLPD Algorithm [L.-Dai-Luo, TSP, 2013]

A New Linear Programming Deflation Algorithm

Step 1. Initialization: Input data $(\mathbf{A}, \mathbf{b}, \bar{\mathbf{p}})$ and set $\mathcal{S} = \mathcal{K}$.

Step 2. Preprocessing (not covered).

Step 3. Power control: Solve linear program (5); check whether all links are supported: if yes, go to **Step 5**; else go to **Step 4**.

Step 4. Admission control: Remove link k_0 according to (6), set $\mathcal{S} = \mathcal{S} \setminus \{k_0\}$, and go to **Step 3**.

Step 5. Postprocessing: Check the removed links for possible admission.

- complexity of solving a LP in form of (5): $O\left(K^{3.5}\right)$
- complexity of the NLPD algorithm: $O\left(K^{4.5}\right)$

L_q Nonconvex Approximation

- L_q $(0 < q < 1)$ nonconvex approximation [L.-Dai-Ma, 2013]:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_q^q + \alpha \bar{\mathbf{p}}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e} \end{aligned}$$

  - $\|\mathbf{x}\|_q^q := \sum_k |[\mathbf{x}]_k|^q$ $(0 < q < 1)$

L_q Nonconvex Approximation

- L_q $(0 < q < 1)$ nonconvex approximation [L.-Dai-Ma, 2013]:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_q^q + \alpha \bar{\mathbf{p}}^T \mathbf{x} \\ \text{s.t.} & \mathbf{0} \leq \mathbf{x} \leq \mathbf{e} \end{array}$$

  - $\|\mathbf{x}\|_q^q := \sum_k |[\mathbf{x}]_k|^q \ (0 < q < 1)$

- Three questions that will be addressed:
  - Why use the nonconvex L_q approximation? Is it better than the convex L_1 approximation? Can the solution of the L_q approximation solve the original sparse problem? $\Leftarrow$ Exact Recovery
  - Is it easy to solve? Is there any polynomial time algorithm which can solve it to global optimality? $\Leftarrow$ Computational Complexity
  - Since the problem is nonconvex, nonsmooth, and non-Lipschitz, how to solve it efficiently? $\Leftarrow$ Algorithm Design

L_1 vs L_q: A Toy Example

- Let $\mathbf{A}, \mathbf{b}, \bar{\mathbf{p}}$ in the JPAC problem (4) be

$$\mathbf{A} = \begin{pmatrix} +1 & 0 & -1 \\ 0 & +1 & -1 \\ -1 & -1 & +1 \end{pmatrix}, \ \mathbf{b} = 0.5\mathbf{e}, \ \bar{\mathbf{p}} = \mathbf{e}$$

- The optimal solution to problem (4) is

$$\mathbf{x}^* = (0.5, 0.5, 0)^T$$

- For any $\alpha \geq 0$, $\mathbf{x} = \mathbf{0}$ is the unique global minimizer of the L_1 approximation problem.
- For any given $q \in (0, 1)$, if α satisfies

$$0 < \alpha < \bar{\alpha}_q := \min\left\{1 + (0.5)^q, 2^q\right\} - (1.5)^q,$$

then the unique global minimizer of the L_q minimization problem is $\mathbf{x}^*$.

Why L_1 Does Not Work Well?

- The problem of minimizing $\|\mathbf{Ax} - \mathbf{b}\|_1$ is equivalent to the problem of minimizing $\|\mathbf{Ax} - \mathbf{b}\|_0$ with high probability under the assumptions that [Candes-Tao, TIT, 2005; Donoho, TIT, 2006]

  1) the vector $\mathbf{Ax} - \mathbf{b}$ at the true solution $\mathbf{x}^*$ is sparse, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $m > n$; and

  2) the entries of the matrix $\mathbf{A}$ is independent and identically distributed (i.i.d.) Gaussian.

- However, these two assumptions often do NOT hold true.

- For instance, $\mathbf{A}$ in the JPAC problem has a special structure, i.e., all diagonal entries are one and all non-diagonal entries are non-positive.

Exact Recovery Result

Theorem (L.-Dai-Ma, 2013)

For any given instance of the JPAC problem (4)*, there exists* $\bar{q} > 0$ *such that when* $q \in (0, \bar{q}]$*, the global solution to its corresponding* L_q *approximation problem is one of the optimal solutions to the JPAC problem* (4)*.*

Theorem (L.-Dai-Ma, 2013)

For any given instance of the JPAC problem (4)*, there exists* $\bar{q} > 0$ *such that when* $q \in (0, \bar{q}]$*, the global solution to its corresponding* L_q *approximation problem is one of the optimal solutions to the JPAC problem* (4)*.*

- This result depends on the special structure of **A** and **b**.
- The $\bar{q}$ is problem-dependent and it is generally not easy to compute it.
- The $\bar{q}$ is generally NOT very small for small networks!
- More works along this direction need to be done...

Complexity Result

Theorem (L.-Dai-Ma, 2013)

For any given $0 < q < 1$, the L_q minimization problem is NP-hard.

- Technical and intricate due to the special structure!

Complexity Result

Theorem (L.-Dai-Ma, 2013)

For any given $0 < q < 1$, the L_q minimization problem is NP-hard.

- Technical and intricate due to the special structure!
- L_q can approximate L_0 BETTER than L_1.
- L_1 is convex while L_q is NP-hard.

Complexity Result

Theorem (L.-Dai-Ma, 2013)

For any given $0 < q < 1$, the L_q minimization problem is NP-hard.

- Technical and intricate due to the special structure!
- L_q can approximate L_0 BETTER than L_1.
- L_1 is convex while L_q is NP-hard.
- What we need is a GOOD algorithm for solving L_q minimization problem!

Reformulation

- For any given $q \in [0, 1]$, the L_q minimization problem is equivalent to

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \|\mathbf{y}\|_q^q + \alpha \bar{\mathbf{p}}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{y} = \mathbf{b},\ \mathbf{x} + \mathbf{z} = \mathbf{e} \\ & \mathbf{x} \geq \mathbf{0},\ \mathbf{y} \geq \mathbf{0},\ \mathbf{z} \geq \mathbf{0} \end{aligned} \tag{8}$$

- Extend the potential reduction algorithm [Ye, MP, 1998; Ge-Jiang-Ye, MP, 2011] to solve problem (8)
  - Potential function:

$$\phi(\mathbf{x},\mathbf{y},\mathbf{z}) = \rho \log\left(\alpha \bar{\mathbf{p}}^T \mathbf{x} + \|\mathbf{y}\|_q^q\right) - \sum_{k=1}^{K} \log\left([\mathbf{x}]_k [\mathbf{y}]_k [\mathbf{z}]_k\right)$$

  - Update rule: the next iterate is chosen as the feasible point that achieves the maximum potential reduction

Why Potential Reduction Algorithms

- Differentiable in the interior feasible region

Why Potential Reduction Algorithms

- Differentiable in the interior feasible region
- Good performance of bypassing local minimizers [Ye, MP, 1998]

Why Potential Reduction Algorithms

- Differentiable in the interior feasible region
- Good performance of bypassing local minimizers [Ye, MP, 1998]

  Run the algorithm with $N > 1$ randomly generated initial points

Why Potential Reduction Algorithms

- Differentiable in the interior feasible region
- Good performance of bypassing local minimizers [Ye, MP, 1998]

  Run the algorithm with $N > 1$ randomly generated initial points
- Polynomial time complexity

Theorem

The interior-point potential reduction algorithm returns an ϵ-KKT point of problem (8) (equivalent to the L_q minimization problem) in no more than

$$O\left(\left(\frac{K^4}{\min\{\epsilon, q\}}\right)\log\left(\frac{1}{\epsilon}\right)\right)$$

operations.

LQMD Algorithm [L.-Dai-Ma, 2013]

The LQMD Algorithm

Step 1. Initialization: Input data $(\mathbf{A}, \mathbf{b}, \bar{\mathbf{p}})$, $q \in (0, 1)$, and positive integer N.

Step 2. Preprocessing (not covered).

Step 3. Power control: Run the potential reduction algorithm N times to solve the L_q minimization problem; check whether all links are supported: if yes, go to **Step 5**; else go to **Step 4**.

Step 4. Admission control: Remove link k_0 according to (6), set $\mathcal{K} = \mathcal{K} \setminus \{k_0\}$, and go to **Step 3**.

Step 5. Postprocessing: Check the removed links for possible admission.

LQMD Algorithm [L.-Dai-Ma, 2013]

The LQMD Algorithm

Step 1. Initialization: Input data $(\mathbf{A}, \mathbf{b}, \bar{\mathbf{p}})$, $q \in (0,1)$, and positive integer N.

Step 2. Preprocessing (not covered).

Step 3. Power control: Run the potential reduction algorithm N times to solve the L_q minimization problem; check whether all links are supported: if yes, go to **Step 5**; else go to **Step 4**.

Step 4. Admission control: Remove link k_0 according to (6), set $\mathcal{K} = \mathcal{K} \setminus \{k_0\}$, and go to **Step 3**.

Step 5. Postprocessing: Check the removed links for possible admission.

- The first nonconvex approximation deflation approach
- The LQMD algorithm: $O\left(\left(\frac{NK^5}{\min\{\epsilon, q\}}\right)\log\left(\frac{1}{\epsilon}\right)\right)$

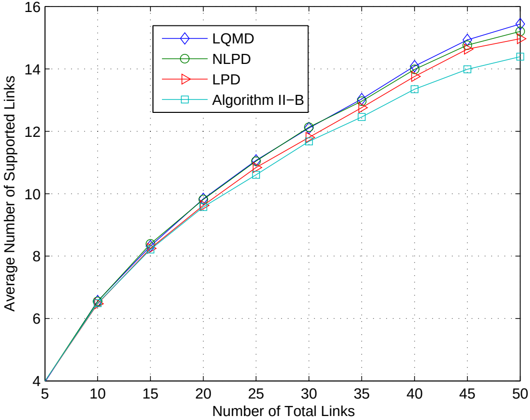
Simulations: Effectiveness of LQMD

- Thank Professor N. D. Sidiropoulos for his help in numerical simulations.
- Compare the proposed LQMD algorithm with other convex approximation deflation algorithms including
  - Algorithm II-B [Mahdavi-Doost-Ebrahimi-Khandani, TIT, 2010]
  - LPD algorithm [Mitliagkas-Sidiropoulos-Swami, TWC, 2011]
  - NLPD algorithm [L.-Dai-Luo, TSP, 2013]
- Comparison criteria
  - number of supported links
  - total transmission power
- All figures report the average results for 200 Monte-Carlo runs.

Simulations: Effectiveness of LQMD

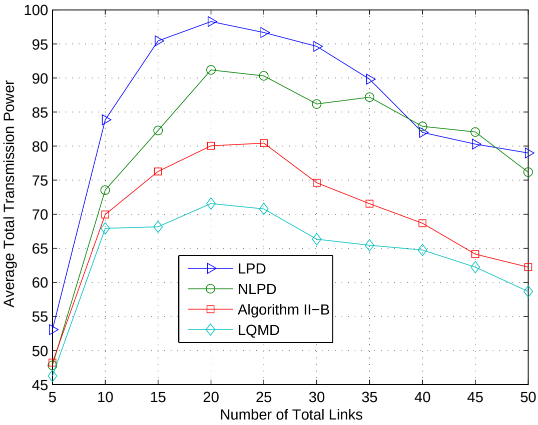
- Thank Professor N. D. Sidiropoulos for his help in numerical simulations.
- Compare the proposed LQMD algorithm with other convex approximation deflation algorithms including
  - Algorithm II-B [Mahdavi-Doost-Ebrahimi-Khandani, TIT, 2010]
  - LPD algorithm [Mitliagkas-Sidiropoulos-Swami, TWC, 2011]
  - NLPD algorithm [L.-Dai-Luo, TSP, 2013]
- Comparison criteria
  - number of supported links
  - total transmission power
- All figures report the average results for 200 Monte-Carlo runs.
- Y.-F. Liu, Y.-H. Dai, and S. Ma, "Joint power and admission control: Non-convex L_q approximation and an effective polynomial time deflation approach."

Average Number of Supported Links



- The proposed LQMD algorithm (with $q = 0.5$ and $N = 5$) supports (slightly) more links than the NLPD algorithm.
- It is shown in [L.-Dai-Luo, TSP, 2013] that the NLPD algorithm can achieve 98% of global optimality in terms of the number of supported links when the number of links is small.

Average Total Transmission Power



- The proposed LQMD algorithm yields SIGNIFICANTLY BETTER total transmission power performance than the NLPD algorithm.
- The proposed LQMD algorithm exhibits very good performance in selecting which subset of links to support.

Concluding Remarks

- Optimal resource allocation in the multi-user interference channel
  - max-min fairness linear transceiver design for a multi-user MIMO IC
  - joint power and admission control for a multi-user SISO IC

Concluding Remarks

- Optimal resource allocation in the multi-user interference channel
  - max-min fairness linear transceiver design for a multi-user MIMO IC
  - joint power and admission control for a multi-user SISO IC
- Complexity theory provides us valuable information in directing our efforts toward those approaches that have the greatest potential of leading to useful algorithms!

Concluding Remarks

- Optimal resource allocation in the multi-user interference channel
  - max-min fairness linear transceiver design for a multi-user MIMO IC
  - joint power and admission control for a multi-user SISO IC
- Complexity theory provides us valuable information in directing our efforts toward those approaches that have the greatest potential of leading to useful algorithms!
- Special structures such as (hidden) convexity, separability, and nonnegativity should be judiciously exploited to design effective/efficient algorithms for optimization problems from real applications!

Concluding Remarks

- Optimal resource allocation in the multi-user interference channel
  - max-min fairness linear transceiver design for a multi-user MIMO IC
  - joint power and admission control for a multi-user SISO IC
- Complexity theory provides us valuable information in directing our efforts toward those approaches that have the greatest potential of leading to useful algorithms!
- Special structures such as (hidden) convexity, separability, and nonnegativity should be judiciously exploited to design effective/efficient algorithms for optimization problems from real applications!
- Application Driven Optimization Is Very Interesting!

Thank You!

Email: yafliu@lsec.cc.ac.cn