INTERFERENCE ALIGNMENT VIA FEASIBLE POINT PURSUIT

Aritra Konar*, Ruoyu Sun†, Nicholas D. Sidiropoulos* and Zhi-Quan Luo*‡
* University of Minnesota, USA, † Stanford University, USA, ‡ Chinese University of Hong Kong, China

UNIVERSITY OF MINNESOTA
Driven to Discover℠

Introduction

- **Goal:** Design an iterative IA algorithm that achieves high DoF for the case of a **constant, *K* user, fully symmetric, generic,** MIMO interference channel, **assuming perfect CSIT.**
- **Contributions:**
  - i. Design of vector space IA scheme is recast as a non-convex quadratic feasibility problem.
  - ii. An extension of the **Feasible Point Pursuit Successive Convex Approximation** (FPP-SCA) [Mehanna *et al.*, 2015] algorithm is proposed to obtain high-quality suboptimal solutions.

System Model

- K users, each with N Tx-Rx antenna pairs, transmitting d streams.
- Estimated signal at k^{th} receiver given by

$$\hat{\mathbf{s}}_k = \mathbf{U}_k^T \mathbf{H}_{kk} \mathbf{V}_k \mathbf{s}_k + \sum_{j=1, j\neq k}^{K} \mathbf{U}_k^T \mathbf{H}_{kj} \mathbf{V}_j \mathbf{s}_j + \mathbf{U}_k^T \mathbf{n}_k$$

- $\mathbf{s}_k \in \mathbb{R}^{d\times 1}$ is the transmitted symbol vector of user k.
- $\mathbf{V}_k \in \mathbb{R}^{N\times d}$ is the linear precoding matrix at k^{th} transmitter.
- $\mathbf{H}_{kj} \in \mathbb{R}^{N\times N}$ is the channel coefficient matrix between transmitter and receiver.
- $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$
- $\mathbf{U}_k \in \mathbb{R}^{N\times d}$ is the post processing matrix at k^{th} receiver.

Problem Description

- For generic channels, problem is equivalent to

$$\begin{aligned} &\underset{\{\mathbf{U}_k,\mathbf{V}_k\}_{k=1}^K}{\text{find}} && \{\mathbf{U}_k,\mathbf{V}_k\}_{k=1}^K && \text{(1a)}\\ &\text{s.t.} && \mathbf{U}_k^T \mathbf{H}_{kj} \mathbf{V}_j = \mathbf{0}, \forall\, j \neq k \in \mathcal{K} && \text{(1b)}\\ & && \text{Rank}(\mathbf{U}_k) = \text{Rank}(\mathbf{V}_k) = d, \forall\, k \in \mathcal{K} && \text{(1c)} \end{aligned}$$

- **Nonconvex, no polynomial-time algorithm known**

Prior Art

- **Leakage Minimization:** distributed algorithm, minimize leakage interference via alternating minimization, low complexity. [Gomadam *et al.*, 2008].
- **MAX SINR:** distributed algorithm, maximize SINR at receivers, low complexity. [Gomadam *et al.*, 2008].
- **Rank Constrained Rank Minimization:** centralized algorithm**,** minimize nuclear norm heuristic via alternating SDPs, high complexity. [Papailiopoulos *et al.*, 2011]

Problem Reformulation

- Reformulate each constraint as a quadratic equality constraint. For example

$$\mathbf{u}_k^{\{l\}T} \mathbf{H}_{k,j} \mathbf{v}_j^{\{m\}} = 0, \forall\, l,m \in \{1,2,\cdots,d\},\ j\neq k \in \mathcal{K}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{u}_k^{\{l\}T} & \mathbf{v}_j^{\{m\}T} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{N\times N} & \mathbf{H}_{k,j} \\ \mathbf{H}_{k,j}^T & \mathbf{0}_{N\times N} \end{bmatrix} \begin{bmatrix} \mathbf{u}_k^{\{l\}} \\ \mathbf{v}_j^{\{m\}} \end{bmatrix} = 0,$$
$$\forall\, l,m \in \{1,2,\cdots,d\},\ j\neq k \in \mathcal{K}$$

$$\Leftrightarrow (\mathbf{x}_{k,j}^{\{l,m\}})^T \mathbf{G}_{k,j} (\mathbf{x}_{k,j}^{\{l,m\}}) = 0, \forall\, l,m \in \{1,2,\cdots,d\},\ j\neq k \in \mathcal{K}$$

- Overall, we have the problem

$$\begin{aligned} &\underset{\{\mathbf{U}_k,\mathbf{V}_k\}_{k=1}^K}{\text{find}} \quad \{\mathbf{U}_k,\mathbf{V}_k\}_{k=1}^K \\ &\text{s.t.} \quad \left.\begin{aligned} \mathbf{x}_{k,j}^{\{l,m\}} &= \begin{bmatrix} \mathbf{u}_k^{\{l\}T} & \mathbf{v}_j^{\{m\}T} \end{bmatrix}^T, \\ (\mathbf{x}_{k,j}^{\{l,m\}})^T \mathbf{G}_{k,j} (\mathbf{x}_{k,j}^{\{l,m\}}) &\gtrless 0 \end{aligned}\right\} \forall\, l,m \in \{1,2,\cdots,d\},\ j\neq k\in\mathcal{K} \quad \text{(Zero forcing constraints)} \\ &\left.\begin{aligned} \|\mathbf{u}_k^{\{l\}}\|_2^2 &\gtrless 1, \\ \|\mathbf{v}_k^{\{l\}}\|_2^2 &\gtrless 1 \end{aligned}\right\} \forall\, l \in \{1,\cdots,d\},\ k\in\mathcal{K} \quad \text{(Unit norm constraints)} \\ &\left.\begin{aligned} \mathbf{y}_k^{\{l,m\}} &= \begin{bmatrix} \mathbf{u}_k^{\{l\}T} & \mathbf{u}_k^{\{m\}T} \end{bmatrix}^T, \\ (\mathbf{y}_k^{\{l,m\}})^T \mathbf{E} (\mathbf{y}_k^{\{l,m\}}) &\gtrless 0 \end{aligned}\right\} \forall\, l\neq m \in \{1,\cdots,d\},\ k\in\mathcal{K} \\ &\text{(Orthogonality constraints)} \\ &\left.\begin{aligned} \mathbf{z}_k^{\{l,m\}} &= \begin{bmatrix} \mathbf{v}_k^{\{l\}T} & \mathbf{v}_k^{\{m\}T} \end{bmatrix}^T, \\ (\mathbf{z}_k^{\{l,m\}})^T \mathbf{E} (\mathbf{z}_k^{\{l,m\}}) &\gtrless 0 \end{aligned}\right\} \forall\, l\neq m \in \{1,\cdots,d\},\ k\in\mathcal{K} \end{aligned}$$

FPP - SCA Algorithm

- **Main Idea:** Use convex inner approximation of feasible set about random starting point and apply positive slacks to maintain feasibility. Impose ℓ_1 penalty on slacks to minimize constraint violations.

$$\begin{aligned} (P_i) \quad &\underset{\{\mathbf{U}_k,\mathbf{V}_k\}_{k=1}^K}{\min.} \sum_{\substack{k,j\in\mathcal{K}, j\neq k,\\ m,l\in\{1,\cdots,d\}}} (s_{k,j}^{\{l,m\}}(1) + s_{k,j}^{\{l,m\}}(2)) \quad \text{(Minimize sum of slacks)} \\ &+ \sum_{\substack{k\in\mathcal{K},\\ l\in\{1,\cdots,d\}}} (s_k^{\{l\}}(1) + s_k^{\{l\}}(2)) \\ &+ \sum_{\substack{k\in\mathcal{K},\\ m\neq l\in\{1,\cdots,d\}}} (s_k^{\{l,m\}}(1) + s_k^{\{l,m\}}(2) + s_k^{\{l,m\}}(3) + s_k^{\{l,m\}}(4)) \\ \text{s.t.} \quad &\left.\begin{aligned} \mathbf{x}_{k,j}^{\{l,m\}} &= \begin{bmatrix} \mathbf{u}_k^{\{l\}T} & \mathbf{v}_j^{\{m\}T} \end{bmatrix}^T, \\ \mathbf{x}_{k,j}^{\{l,m\}} &\in \mathcal{C}_{\text{ZF}}(\mathbf{u}_k^{\{l\}}, \mathbf{v}_j^{\{m\}}), \\ s_{k,j}^{\{l,m\}}(1) \geq 0,&\ s_{k,j}^{\{l,m\}}(2) \geq 0 \end{aligned}\right\} \forall\, l,m \in \{1,2,\cdots,d\},\ j\neq k\in\mathcal{K} \quad \text{(Zero forcing constraints)} \\ &\left.\begin{aligned} \mathbf{u}_k^{\{l\}} &\in \mathcal{C}_{\text{UN}}(\mathbf{u}_k^{\{l\}}), \\ \mathbf{v}_k^{\{l\}} &\in \mathcal{C}_{\text{UN}}(\mathbf{v}_k^{\{l\}}), \\ s_k^{\{l\}}(1) \geq 0,&\ s_k^{\{l\}}(2) \geq 0 \end{aligned}\right\} \forall\, l \in \{1,\cdots,d\}, k\in\mathcal{K} \quad \text{(Unit norm constraints)} \\ &\left.\begin{aligned} \mathbf{y}_k^{\{l,m\}} &= \begin{bmatrix} \mathbf{u}_k^{\{l\}T} & \mathbf{u}_k^{\{m\}T} \end{bmatrix}^T, \\ \mathbf{y}_k^{\{l,m\}} &\in \mathcal{C}_{\text{ORTH}}(\mathbf{u}_k^{\{l\}}, \mathbf{u}_k^{\{m\}}), \\ s_k^{\{l,m\}}(1) \geq 0,&\ s_k^{\{l,m\}}(2) \geq 0 \end{aligned}\right\} \forall\, l\neq m \in \{1,\cdots,d\},\ k\in\mathcal{K} \\ &\text{(Orthogonality constraints)} \\ &\left.\begin{aligned} \mathbf{z}_k^{\{l,m\}} &= \begin{bmatrix} \mathbf{v}_k^{\{l\}T} & \mathbf{v}_k^{\{m\}T} \end{bmatrix}^T, \\ \mathbf{z}_k^{\{l,m\}} &\in \mathcal{C}_{\text{ORTH}}(\mathbf{v}_k^{\{l\}}, \mathbf{v}_k^{\{m\}}), \\ s_k^{\{l,m\}}(3) \geq 0,&\ s_k^{\{l,m\}}(4) \geq 0 \end{aligned}\right\} \forall\, l\neq m \in \{1,\cdots,d\},\ k\in\mathcal{K} \end{aligned}$$

Overall Algorithm

Step 0: Randomly generate initialization points $\{\mathbf{p}_{k,j}^{\{l,m\}(0)}\}$, $\{\mathbf{q}_k^{\{l\}(0)}, \mathbf{r}_k^{\{l\}(0)}\}$, $\{\mathbf{t}_k^{\{l,m\}(0)}, \mathbf{w}_k^{\{l,m\}(0)}\}$ for each pair of constraints in (P_0).

Step i: Solve the problem (P_i) to obtain a set of solutions $\{\mathbf{U}_k^{(i)}, \mathbf{V}_k^{(i)}\}_{k=1}^K$.

Update $\{\mathbf{p}_{k,j}^{\{l,m\}(i+1)} := \begin{bmatrix} \mathbf{u}_k^{\{l\}(i)T} & \mathbf{v}_j^{\{m\}(i)T} \end{bmatrix}^T\}$;

$\{\mathbf{q}_k^{\{l\}(i+1)}\} := \mathbf{u}_k^{\{l\}(i)}$; $\{\mathbf{r}_k^{\{l\}(i+1)},\} := \mathbf{v}_k^{\{l\}(i)}$;

$\{\mathbf{t}_k^{\{l,m\}(i+1)} := \begin{bmatrix} \mathbf{u}_k^{\{l\}(i)^T} & \mathbf{u}_k^{\{m\}(i)^T} \end{bmatrix}^T\}$;

$\{\mathbf{w}_k^{\{l,m\}(i+1)} := \begin{bmatrix} \mathbf{v}_k^{\{l\}(i)^T} & \mathbf{v}_k^{\{m\}(i)^T} \end{bmatrix}^T\}$;

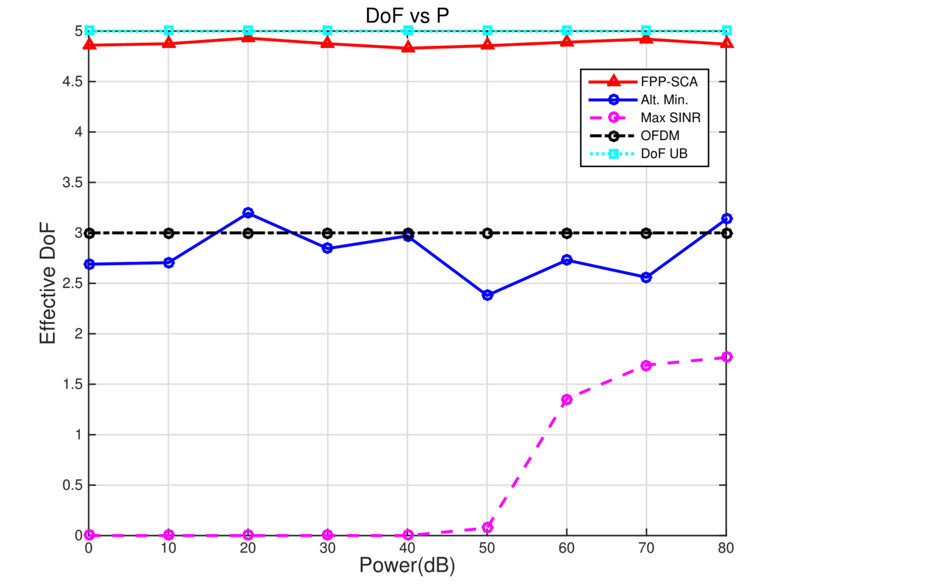
Repeat until feasibilty achieved or maximum number of iterations

- Generates a monotonically non-increasing, convergent cost sequence.
- If cost function converges to zero, a solution of problem (1) will be obtained.

Simulation Results

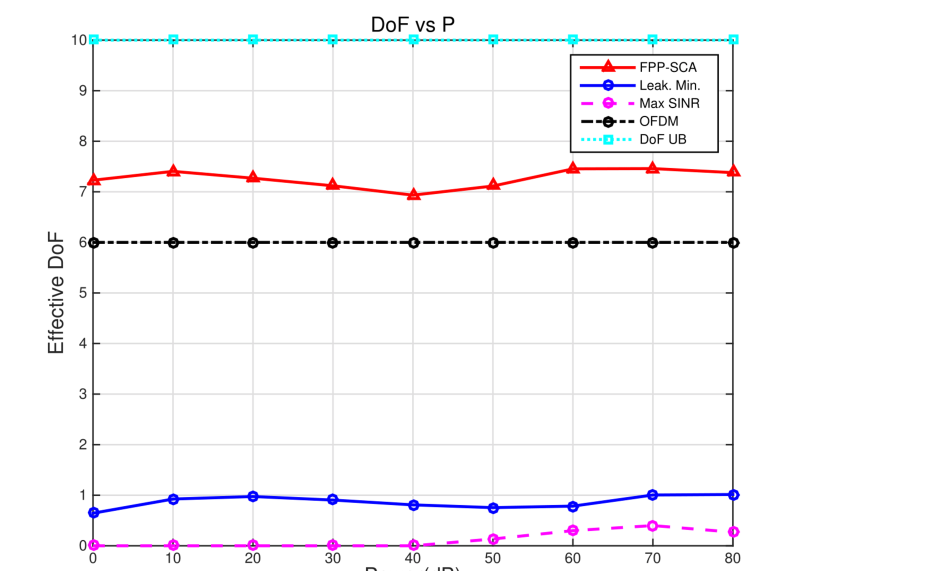
- FPP-SCA run for a maximum of 120 iterations, with a maximum of 5 restarts. Leakage minimization and Max SINR run for maximum of 10^4 iterations.

Fig 1: DoF for a $(3\times 3, 1)^5$ MIMO IC



Performance of FPP-SCA very close to theoretical upper bound. Other algorithms significantly worse in comparison.

Fig 2: DoF for a $(6\times 6, 2)^5$ MIMO IC



FPP-SCA achieves significantly higher DoF in comparison to other algorithms..

Future Work:

- Low complexity version of FPP-SCA.
- Extension to more general channel settings, e.g., SISO IC with generic channel extensions.