Deep Weighted MMSE Downlink Beamforming

Lissy Pellaco, Mats Bengtsson, Joakim Jaldén

KTH Royal Institute of Technology, Stockholm, Sweden

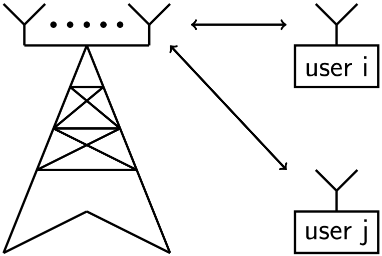
IEEE ICASSP, 6-11 June, 2021

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel

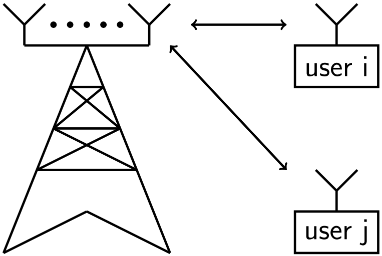
Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users



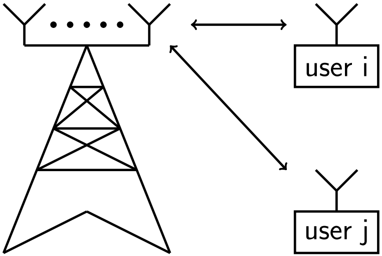
Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Introduction

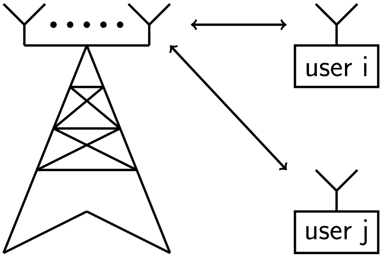
- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Goals

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming

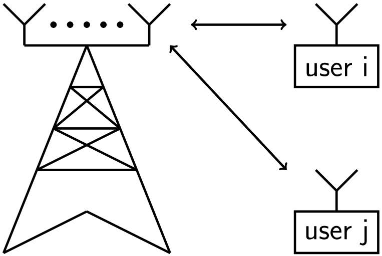


Goals

1. Maximize **weighted sum rate** subject to **transmit power constraint**

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming

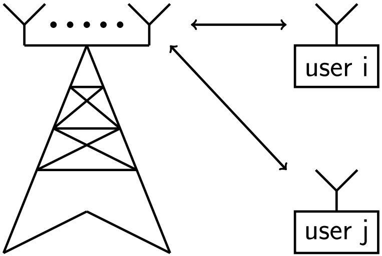


Goals

1. Maximize **weighted sum rate** subject to **transmit power constraint**
2. Satisfy the **power consumption** and **latency** requirements at the base station

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming

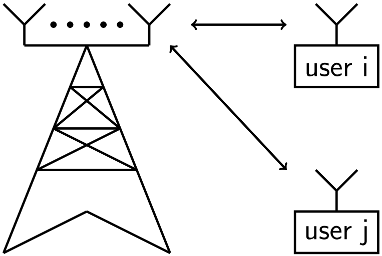


Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j \neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



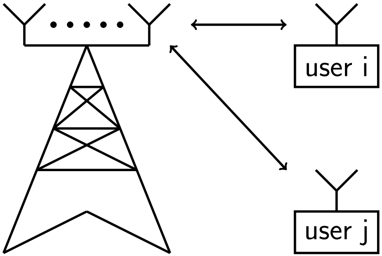
Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

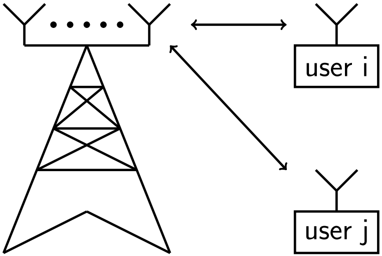
$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j \neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

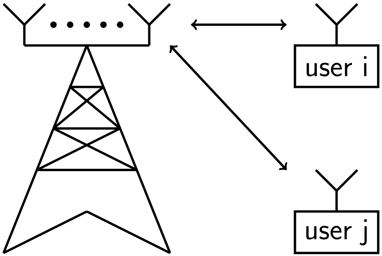
$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

$x_i \sim \mathcal{CN}(0, 1)$: transmitted symbol

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

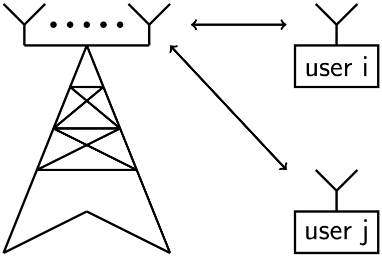
$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

$x_i \sim \mathcal{CN}(0, 1)$: transmitted symbol

$n_i \sim \mathcal{CN}(0, \sigma^2)$: noise

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

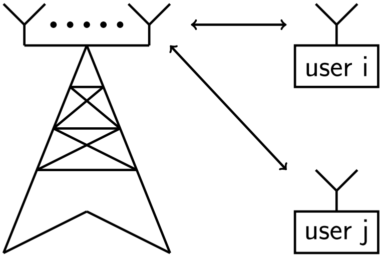
$x_i \sim \mathcal{CN}(0, 1)$: transmitted symbol

$n_i \sim \mathcal{CN}(0, \sigma^2)$: noise

$$\boldsymbol{V} \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]^T \in \mathbb{C}^{N\times M}$$

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

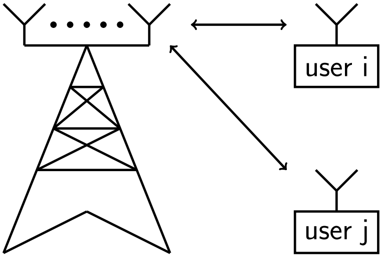
$x_i \sim \mathcal{CN}(0, 1)$: transmitted symbol

$n_i \sim \mathcal{CN}(0, \sigma^2)$: noise

$$\boldsymbol{V} \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]^T \in \mathbb{C}^{N\times M}$$

Introduction

- Multi-user multiple-input single-output (MU-MISO) interference downlink channel
- Single base station with M transmit antennas, N single-antenna users
- Linear beamforming



Received signal

$$y_i = \boldsymbol{h}_i^H \boldsymbol{v}_i x_i + \sum_{j=1, j\neq i}^{N} \boldsymbol{h}_i^H \boldsymbol{v}_j x_j + n_i$$

$\boldsymbol{h}_i \in \mathbb{C}^M$: channel

$\boldsymbol{v}_i \in \mathbb{C}^M$: beamformer vector

$x_i \sim \mathcal{CN}(0, 1)$: transmitted symbol

$n_i \sim \mathcal{CN}(0, \sigma^2)$: noise

$$\boldsymbol{V} \triangleq [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]^T \in \mathbb{C}^{N\times M}$$

Beamformer matrix

Beamforming problem

- We address the weighted sum rate (WSR) maximization problem

$$\max_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \alpha_i \log_2 (1 + \mathrm{SINR}_i) \tag{1a}$$

$$\text{s.t.} \quad \mathtt{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{1b}$$

Beamforming problem

- We address the weighted sum rate (WSR) maximization problem

$$\max_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \alpha_i \log_2 \left(1 + \mathrm{SINR}_i\right) \tag{1a}$$

$$\text{s.t.} \quad \mathtt{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{1b}$$

  - $\log_2 (1 + \mathrm{SINR}_i)$ is the rate of user i

Beamforming problem

- We address the weighted sum rate (WSR) maximization problem

$$\max_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \alpha_i \log_2\left(1 + \mathrm{SINR}_i\right) \tag{1a}$$

$$\text{s.t.} \quad \mathtt{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{1b}$$

  - $\log_2\left(1 + \mathrm{SINR}_i\right)$ is the rate of user i
  - $\mathrm{SINR}_i = \frac{|\boldsymbol{h}_i^H \boldsymbol{v}_i|^2}{\sum_{j=1, j\neq i}^{N} |\boldsymbol{h}_i^H \boldsymbol{v}_j|^2 + \sigma^2}$ (signal-to-interference-plus-noise ratio of user i)

Beamforming problem

- We address the weighted sum rate (WSR) maximization problem

$$\max_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \alpha_i \log_2 \left(1 + \mathrm{SINR}_i\right) \tag{1a}$$

$$\text{s.t.} \quad \mathtt{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{1b}$$

  - $\log_2 (1 + \mathrm{SINR}_i)$ is the rate of user i
  - $\mathrm{SINR}_i = \frac{|\boldsymbol{h}_i^H \boldsymbol{v}_i|^2}{\sum_{j=1, j\neq i}^{N} |\boldsymbol{h}_i^H \boldsymbol{v}_j|^2 + \sigma^2}$ (signal-to-interference-plus-noise ratio of user i)
  - α_i is the priority of user i (assumed to be known)

Beamforming problem

- We address the weighted sum rate (WSR) maximization problem

$$\max_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \alpha_i \log_2(1 + \mathrm{SINR}_i) \tag{1a}$$

$$\text{s.t.} \quad \mathtt{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{1b}$$

  - $\log_2(1 + \mathrm{SINR}_i)$ is the rate of user i
  - $\mathrm{SINR}_i = \frac{|\boldsymbol{h}_i^H \boldsymbol{v}_i|^2}{\sum_{j=1, j\neq i}^{N} |\boldsymbol{h}_i^H \boldsymbol{v}_j|^2 + \sigma^2}$ (signal-to-interference-plus-noise ratio of user i)
  - α_i is the priority of user i (assumed to be known)

- Problem (1) is known to be **NP-hard**[1]

[1] Luo et al., "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, 2008.

WMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

- At each iteration of the WMMSE:

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

- At each iteration of the WMMSE:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi},\boldsymbol{w},\boldsymbol{V})$

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \le P \tag{2b}$$

- At each iteration of the WMMSE:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi},\boldsymbol{w},\boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{V})$

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

- At each iteration of the WMMSE:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi},\boldsymbol{w},\boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{V})$
  - the update of $\boldsymbol{V}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\xi})$ s.t. $\mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P$

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

- At each iteration of the WMMSE:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi}, \boldsymbol{w}, \boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{V})$
  - the update of $\boldsymbol{V}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi})$ s.t. $\mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P$
- It is **guaranteed to converge** to a local optimum

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

WMMSE algorithm

- The weighted minimum mean square error (WMMSE) algorithm[2] finds a **local optimum** by working on an **equivalent reformulation** of the WSR problem, i.e.,

$$\min_{\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}} \quad f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{V}) \tag{2a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{V}\boldsymbol{V}^H) \leq P \tag{2b}$$

- At each iteration of the WMMSE:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi},\boldsymbol{w},\boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u},\boldsymbol{\xi},\boldsymbol{V})$
  - the update of $\boldsymbol{V}$ is the optimal solution of $\min_{\boldsymbol{\xi}} f(\boldsymbol{u},\boldsymbol{w},\boldsymbol{\xi})$ s.t. $\mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P$
- It is **guaranteed to converge** to a local optimum
- It exhibits a **relatively high computational complexity**

[2] Shi et al.,"An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transaction on Signal Processing*, 2011.

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm
  - Map each iteration of the algorithm to a neural network layer

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm
  - Map each iteration of the algorithm to a neural network layer
  - Fix the number of layers of the network according to the complexity and latency constraints

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm
  - Map each iteration of the algorithm to a neural network layer
  - Fix the number of layers of the network according to the complexity and latency constraints
  - Select the trainable parameters

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm
  - Map each iteration of the algorithm to a neural network layer
  - Fix the number of layers of the network according to the complexity and latency constraints
  - Select the trainable parameters
  - Train the network with gradient-based methods and back-propagation

Deep unfolding

- It is a learning technique applicable to **iterative algorithms**
- **Goal**: trade off complexity and performance in presence of constraints
- **Key idea**: build and train a neural network whose structure is determined by the iterative algorithm
  - Map each iteration of the algorithm to a neural network layer
  - Fix the number of layers of the network according to the complexity and latency constraints
  - Select the trainable parameters
  - Train the network with gradient-based methods and back-propagation
- It incorporates **domain knowledge** in the structure of the network

Deep unfolding

Advantages with respect to standard neural network solutions

- No architecture selection
- Explainability
- Fewer parameters to train

WMMSE algorithm - Deep unfolding

- The WMMSE algorithm involves operations that are **hard to map to neural network layers as acknowledged by Sun et al.**[3]

[3] Sun et al., "Learning to Optimize: Training Deep Neural Networks for Interference Management," *IEEE Transactions on Signal Processing*, 2018

WMMSE algorithm - Deep unfolding

- The WMMSE algorithm involves operations that are **hard to map to neural network layers as acknowledged by Sun et al.**[3]

WMMSE steps	Unfolded steps
$\boldsymbol{u}_j = \text{argmin}_{\xi} f(\xi, \boldsymbol{w}_{j-1}, \boldsymbol{V}_{j-1})$	$\boldsymbol{u}_j = \Omega(\boldsymbol{w}_{j-1}, \boldsymbol{V}_{j-1})$
$\boldsymbol{w}_j = \text{argmin}_{\xi} f(\boldsymbol{u}_j, \xi, \boldsymbol{V}_{j-1})$	$\boldsymbol{w}_j = \Psi(\boldsymbol{u}_j, \boldsymbol{V}_{j-1})$
$\boldsymbol{V}_j = \text{argmin}_{\xi} f(\boldsymbol{u}_j, \boldsymbol{w}_j, \xi)$ s.t. $\text{Tr}(\xi\xi^H) \leq P$	**?**

j is the iteration index

[3] Sun et al., "Learning to Optimize: Training Deep Neural Networks for Interference Management," *IEEE Transactions on Signal Processing*, 2018

WMMSE algorithm - Deep unfolding

- The update equation of $\boldsymbol{V}$ is obtained by solving

$$\min_{\boldsymbol{\xi}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi}) \tag{3a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P, \tag{3b}$$

with the method of Lagrange multipliers

WMMSE algorithm - Deep unfolding

- The update equation of $\boldsymbol{V}$ is obtained by solving

$$\min_{\boldsymbol{\xi}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi}) \tag{3a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P, \tag{3b}$$

  with the method of Lagrange multipliers
- It leads to **a matrix inversion, an eigendecomposition, and a bisection search**

WMMSE algorithm - Deep unfolding

- The update equation of $\boldsymbol{V}$ is obtained by solving

$$\min_{\boldsymbol{\xi}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi}) \tag{3a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P, \tag{3b}$$

  with the method of Lagrange multipliers
- It leads to **a matrix inversion, an eigendecomposition, and a bisection search**
- We observe that
  - The cost function is convex
  - The constraint set is convex

WMMSE algorithm - Deep unfolding

- The update equation of $\boldsymbol{V}$ is obtained by solving

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi}) && (3a) \\ \text{s.t.} \quad & \mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \le P, && (3b) \end{aligned}$$

  with the method of Lagrange multipliers
- It leads to **a matrix inversion, an eigendecomposition, and a bisection search**
- We observe that
  - The cost function is convex
  - The constraint set is convex
- We propose to solve (3) with the **projected gradient descent (PGD)** approach

WMMSE algorithm - Deep unfolding

- The update equation of $\boldsymbol{V}$ is obtained by solving

$$\min_{\boldsymbol{\xi}} \quad f(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\xi}) \tag{3a}$$

$$\text{s.t.} \quad \mathrm{Tr}(\boldsymbol{\xi}\boldsymbol{\xi}^H) \leq P, \tag{3b}$$

  with the method of Lagrange multipliers
- It leads to **a matrix inversion, an eigendecomposition, and a bisection search**
- We observe that
  - The cost function is convex
  - The constraint set is convex
- We propose to solve (3) with the **projected gradient descent (PGD)** approach
- We truncate the sequence of PGD steps to K

Unfoldable WMMSE algorithm

- At each iteration:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\xi} f(\xi, \boldsymbol{w}, \boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\xi} f(\boldsymbol{u}, \xi, \boldsymbol{V})$
  - the update of $\boldsymbol{V}$ is given by K PGD steps

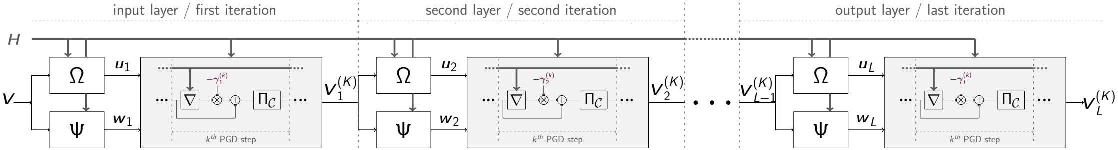
Unfoldable WMMSE algorithm

- At each iteration:
  - the update of $\boldsymbol{u}$ is the optimal solution of $\min_{\xi} f(\xi, \boldsymbol{w}, \boldsymbol{V})$
  - the update of $\boldsymbol{w}$ is the optimal solution of $\min_{\xi} f(\boldsymbol{u}, \xi, \boldsymbol{V})$
  - the update of $\boldsymbol{V}$ is given by K PGD steps

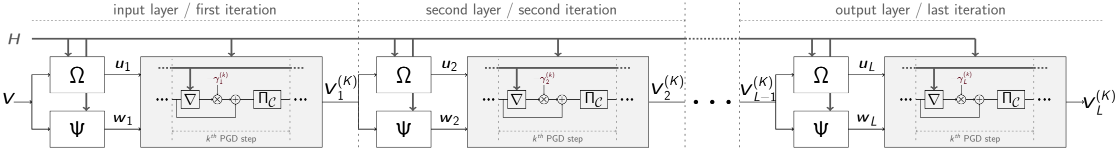
Convergence

We can prove that the unfoldable WMMSE algorithm retains the **same convergence guarantees** of the original WMMSE

Deep unfolded WMMSE

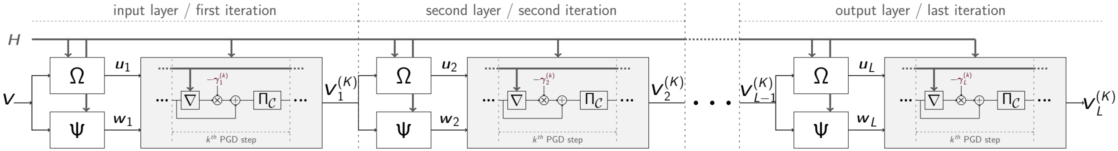


Deep unfolded WMMSE



- We select the step sizes of the PGD (Γ) to be the trainable parameters

Deep unfolded WMMSE

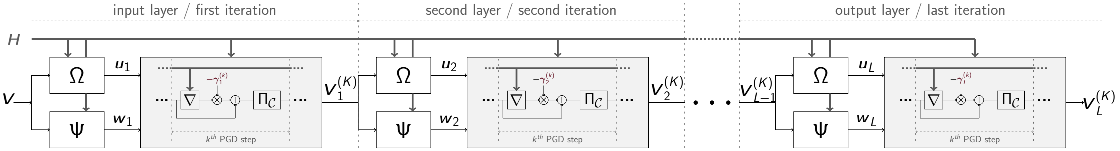


- We select the step sizes of the PGD (Γ) to be the trainable parameters
- We minimize the following loss function

$$\mathcal{L}(\Gamma) = -\frac{1}{N_\mathrm{s}} \sum_{n=1}^{N_\mathrm{s}} \sum_{l=1}^{L} f_\mathrm{WSR}(\boldsymbol{H}_n, \boldsymbol{V}_l; \Gamma) \tag{4}$$

where N_s is the size of the training set

Deep unfolded WMMSE



- We select the step sizes of the PGD (Γ) to be the trainable parameters
- We minimize the following loss function

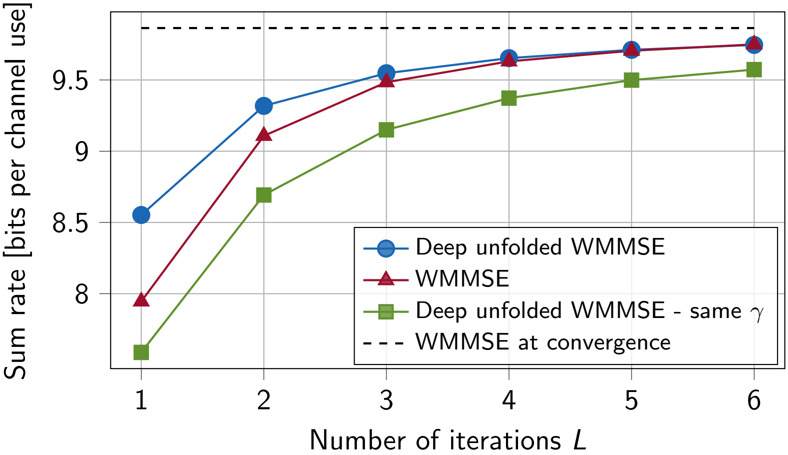
$$\mathcal{L}(\Gamma) = -\frac{1}{N_\mathrm{s}} \sum_{n=1}^{N_\mathrm{s}} \sum_{l=1}^{L} \underline{f_{\mathrm{WSR}}}(\boldsymbol{H}_n, \boldsymbol{V}_l; \Gamma) \tag{4}$$

Weighted Sum Rate

where N_s is the size of the training set

Numerical results

- $M = 4$
- $N = 4$
- $\frac{P}{\sigma^2} = 10$ dB
- 4 PGD steps



Conclusion

- We addressed the **trade-off between complexity and performance** for the WSR maximization beamforming problem

Conclusion

- We addressed the **trade-off between complexity and performance** for the WSR maximization beamforming problem
- To this end, we provided a **variant of the WMMSE** algorithm that
  - allows for the novel application of deep unfolding
  - retains the same convergence guarantees of the original WMMSE algorithm

Conclusion

- We addressed the **trade-off between complexity and performance** for the WSR maximization beamforming problem
- To this end, we provided a **variant of the WMMSE** algorithm that
  - allows for the novel application of deep unfolding
  - retains the same convergence guarantees of the original WMMSE algorithm
- **Numerical results** confirmed that the deep unfolded WMMSE successfully addresses the trade-off

Thank you for your attention!

https://github.com/lpkg/ WMMSE-deep-unfolding/tree/ICASSP2021

You can reach out to me at **pellaco@kth.se**