Low Complexity Multiuser MIMO Scheduling with Channel Decomposition

Jungwoo Lee
Wireless Signal Processing Lab
School of Electrical Engineering
Seoul National University
junglee@ee.snu.ac.kr
Contents

- Introduction
- Precoding Techniques
  - Channel Decomposition
- Low Complexity Scheduling Algorithms
  - Max-upperbound Scheduling Algorithms
- Simulation Results
- Conclusions
Introduction

- Single-user MIMO System
  - Coordination among receive antennas

- Multiuser MIMO System
  - Receive antennas of different users not able to coordinate → Multiuser interference
  - Precoding techniques with full CSI

- Scheduling Algorithms
  - BS selects a subset of users to serve
    → Maximize the total throughput
  - Optimal exhaustive search algorithm
  - Low complexity scheduling algorithm
Channel Decomposition (1)

- **System model**

  Received signal of user $k$ is

  $$ r_k = H_k \sum_{i=1}^{K} T_i s_i + n_k = H_k T_k s_k + H_k \sum_{i=1, i \neq k}^{K} T_i s_i + n_k $$

  Precoding matrix
Channel Decomposition (2)

- **How to Find $T_k$?**

\[
\begin{align*}
H_1 \sum_{i=1, i \neq 1}^{K} T_i s_i &= 0 \\
H_2 \sum_{i=1, i \neq 2}^{K} T_i s_i &= 0 \\
M \sum_{i=1, i \neq K}^{K} T_i s_i &= 0 \\
\end{align*}
\]

s.t. \(\sum_{k=1}^{K} \text{trace}(T_k T_k^H) = P_T\)

\[T_k = 0 \quad \text{for} \quad k = 1, K, K.\]

- to guarantee the existence of nonzero solution

\[M > \max \left\{ \sum_{i=1, i \neq k}^{K} N_i, \quad k = 1, 2, ..., K \right\}\]

- use SVD to get an orthonormal basis of \(\text{Null}(H_i)\)

denoted as \(\{q_k(1), q_k(2), \ldots, q_k(n_k)\}\)

\[n_k = M - \sum_{i=1, i \neq k}^{K} N_i\]
Channel Decomposition (3)

- Denote $Q_k = [q_k(1) q_k(2) \ldots q_k(n_k)]$, and $B_k$ is the transmit processing matrix, thus $T_k = Q_k B_k$.

- Received signal of user $k$ becomes

$$r_k = H_k Q_k B_k s_k + n_k$$

- The channel after decomposition

![Diagram](image-url)
Scheduling Algorithms

- Assumptions
  - All MSs have same number of receive antennas
  - Utilize all receive antennas when scheduled for transmission

- Why we need user scheduling?
  - Large number of users
  - Number of simultaneously supportable users with channel decomposition ($K$) is limited
  - Select users to maximize system throughput

User Diversity
Optimal Scheduling Algorithm (1)

- Channel capacity when transmitter knows $\mathbf{H}$
  
  $C(\mathbf{H}, P/N_0) = \sum_{i=1}^{r} \log_2 \left( 1 + \frac{P \gamma_i}{N_0 M} \lambda_i \right)$

  where $\lambda_i$’s are eigenvalues of $\mathbf{H}^H \mathbf{H}$ and $\gamma_i = \left( \mu - \frac{N_0 M}{P \lambda_i} \right)^+$, $\sum_{i=1}^{n} \gamma_i = M$

- Equal power allocation among users
  
  For certain user set $S$,

  $$R_{CD}^{EP}(S) = \sum_{k=1}^{K} C \left( \mathbf{H}_k \mathbf{Q}_k, \frac{P}{N_0 K} \right)$$

  → Optimal scheduling algorithm

  $$R_{CD}^{EP} = \max_{S \subset \{1, K, K_T\} ; |S| \leq K} R_{CD}^{EP}(S)$$
Optimal Scheduling Algorithm (2)

- Require an exhaustive search over the entire user set
  - Each possible choice of user group, system throughput is calculated

- Search space of optimal algorithm:
  - number of possible user group
    \[ \sum_{i=1}^{K} \binom{K_T}{i} \]
  - when number of total users \( K_T \) is large, the search space becomes prohibitively huge
  - for example, \( K_T = 40 \) and \( K = 4 \) \[ \sum_{i=1}^{K} \binom{K_T}{i} \approx 200,000 \]
Low Complexity Scheduling Algorithm (1)

- An upper bound to the sum rate of multiuser MIMO system can be obtained by letting the receivers cooperate.
- Capacity equation of single user MIMO:
  \[ C = \log_2 \det \left( I_M + \frac{P}{N_0 M} \mathbf{H}^H \mathbf{H} \right) \]
  - upper bound of multiuser MIMO
  - Tx in BS  →  Column number of \( \mathbf{H} \)
  - Rx in all scheduled users  →  Row number of \( \mathbf{H} \)
Low Complexity Scheduling Algorithm (2)

- Adding user $n$ to user set $S$, the capacity upper bound can be written as:

$$C\left(\begin{bmatrix} H(S) \\ H_n \end{bmatrix}\right) = \log_2 \det\left( I_M + \frac{P}{N_0 M} H(S)^H H(S) \right)$$

$$+ \log_2 \det\left( I_N + \frac{P}{N_0 M} H_n \left( I_M + \frac{P}{N_0 M} H(S)^H H(S) \right)^{-1} H_n^H \right)$$

- Maximize the capacity:

$$s_{k+1} = \arg \max_{n \in S} \det\left( I_N + H_n \left( \frac{N_0 M}{P} I_M + H(S)^H H(S) \right)^{-1} H_n^H \right)$$

- Inversion can be computed recursively with matrix inversion lemma:

$$\left( A + B^H B \right)^{-1} = A^{-1} - A^{-1} B^H \left( I_N + BA^{-1} B^H \right)^{-1} BA^{-1}$$
Low Complexity Scheduling Algorithm (3)

- Max-upperbound Scheduling Algorithm

**Step I: Initialization:**

- \( T = \{1, 2, K, K_r\} \); 
- \( S = \emptyset \); 
- \( R_{last} = 0 \); 
- \( A = \frac{P}{N_0M} I_M \); 

**Step II: Loop**

FOR \( i = 1 \) to \( K \)

- \( p = \arg \max_{t \in T} \det \left( I_N + H_t A H_t^H \right) \); 
- \( R_{temp} = R_{CD}^{EP} (S_{temp}) \), where \( S_{temp} = S + \{p\} \); 
- IF \( R_{temp} < R_{last} \)

  selected user set is \( S \), algorithm ends

ELSE

- \( S = S_{temp} \); \( R_{last} = R_{temp} \); \( T = T - \{p\} \); 

END-IF

- \( A = A - A H_p^H \left( I_N + H_p A H_p^H \right)^{-1} H_p A \); 

END-FOR
Low Complexity Scheduling Algorithm (3)

- **Scheduling Procedure**
  - Firstly, select one user with the **highest capacity**
  - Then, add one user to the selected user group in each iteration, the user provide the **highest capacity upper bound** with those selected users
  - Iteration terminates when $K$ users are selected or the total throughput decreases after more user being added
  - More users selected → the decomposed channels lose the transmit diversity in order to cancel the multiuser interference
  - The optimal number of users **varies** for different system configuration and **SNR**
Referred Algorithms in Simulations (1)

- Capacity-Based Suboptimal User Selection Algorithm [Shen ‘05]

1) Initially, let $\Omega = \{1, 2, \cdots, K\}$ and $\Upsilon = \emptyset$. Let $s_1 = \arg\max_{k \in \Omega} \log \left| I + \frac{1}{\sigma^2} H_k Q_k H_k^* \right|$ where $\text{Tr}(Q_k) \leq P$ and $Q_k$ is semi-positive definite. Let $\Omega = \Omega - \{s_1\}$ and $\Upsilon = \Upsilon + \{s_1\}$. Let $C_{temp} = \max_{k \in \Omega} \log \left| I + \frac{1}{\sigma^2} H_k Q_k H_k^* \right|$.

2) for $i = 2 : K$

   a) For every $k \in \Omega$,

      i) Let $\bar{\Upsilon}_k = \Upsilon + \{k\}$.

      ii) Find the precoding matrix $T_j$ for each $j \in \bar{\Upsilon}_k$, and obtain the effective channel $\bar{H}_j = H_j T_j$ for each $j \in \bar{\Upsilon}_k$.

      iii) Perform a singular value decomposition (SVD) on $\bar{H}_j$, and obtain the $M$ singular values $\{\lambda_{j,m}\}_{i=1}^M$.

      iv) Water-fill over $\lambda_{j,m}^2$ for $j \in \bar{\Upsilon}_k$ and $1 \leq m \leq M$. Find the total throughput to the user set $\bar{\Upsilon}_k$, denoted as $C_k$.

   b) Let $s_i = \arg\max_{k \in \Omega} C_k$.

   c) If $\max_{k \in \Omega} C_k < C_{temp}$

      Algorithm terminated. The selected user set is $\Upsilon$.

      else

      Let $\Omega = \Omega - \{s_i\}$ and $\Upsilon = \Upsilon + \{s_i\}$. And let $C_{temp} = \max_{k \in \Omega} C_k$. 


Frobenius Norm-Based Suboptimal User Selection Algorithm [Shen ‘05]

1) Initially, let $\Omega = \{1, 2, \ldots, K\}$ and $\Upsilon = \emptyset$. Let $s_1 = \arg \max_{k \in \Omega} ||H_k||^2_F$. Let $V = V_{s_1}$. Let $\Omega = \Omega - \{s_1\}$ and $\Upsilon = \Upsilon + \{s_1\}$.

2) for $i = 2 : \hat{K}$

   a) For each $k \in \Omega$, let $\tilde{H}_k = H_k - H_k V^* V$. Then $\tilde{H}_k$ is in the null space of $V$.
      for $j = 1 : i - 1$
      i) Let
      \[
      \tilde{H}_{s_{j,k}} = [H_{s_{j+1}}^T \cdots H_{s_{j-1}}^T H_{s_{j-1}}^T \cdots H_{s_{k}}^T H_{s_{k}}^T]^T.
      \]

      ii) Let $W_{s_{j,k}}$ be the row basis for $\tilde{H}_{s_{j,k}}$ after Gram-Schmidt orthogonalization.

   b) For each $s \in \Upsilon$, let $\tilde{H}_s = H_s - H_s W_{s,k}^* W_{s,k}$. Then $\tilde{H}_s$ is in the null space of $\tilde{H}_{s,k}$. Let
      \[
      s_i = \arg \max_{k \in \Omega} \left( \sum_{s \in \Upsilon} ||\tilde{H}_s||^2_F + ||\tilde{H}_k||^2_F \right).
      \]

   c) Let $\Omega = \Omega - \{s_i\}$ and $\Upsilon = \Upsilon + \{s_i\}$. Apply the Gram-Schmidt orthogonalization procedure to $H_{s_i}$ and get $\tilde{V}_{s_i}$. Let $V = [V^T \tilde{V}_{s_i}^T]^T$.

3) Apply the capacity-based suboptimal user selection algorithm to the set $\Upsilon$, and get the final selected user set and the total throughput.
Simulation Results (1)

- Sum-rate capacity comparison of various scheduling algorithms

  - 8 transmit antennas at the base station and 4 receive antennas at each user

---

![Graph showing sum-rate capacity comparison](image)
Simulation Results (2)

- Sum-rate capacity comparison of various scheduling algorithms

- 8 transmit antennas at the base station and 4 receive antennas at each user

![Graph showing sum-rate capacity comparison](image-url)

- Optimal
- Capacity-based
- Proposed algorithm
- Frobenius norm

Number of total users vs. Sum-rate Capacity (bits/s/Hz) for different SNR levels (20 dB, 10 dB, 0 dB).
Simulation Results (3)

- Computational Complexity comparison
  - Average CPU run time comparison for various scheduling algorithms with Tx-8 and Rx-2
Conclusions

- Precoding techniques decompose the multiuser MIMO downlink channel into parallel independent single-user MIMO channels

- Compared with other two low complexity scheduling algorithms, the proposed algorithm can achieve almost the same (or better) throughput, but with much lower computational complexity
  - In compared algorithms, calculation of SVD of channel matrix, matrix Frobenius norm and Gram-Schmidt orthogonalization is required frequently
  - In proposed algorithm, inversion and determinant of an $N \times N$ matrix need been calculated, where $N$ is the number of receive antennas in MS which is usually very small (1 or 2)