Clustering for Interference Alignment in a Multiuser Interference Channel

Sujie Chen and Roger S. Cheng
Department of Electronic & Computer Engineering
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
Email: {eechsj, eecheng}@ust.hk

***Abstract*—Interference alignment (IA) has been shown as a promising technique to achieve the optimal capacity scaling of a multiuser interference channel at asymptotically high signal-to-noise-ratio (SNR). However, in practical communication systems, mitigating all interference via IA is not necessary since some of them only have negligible effect due to path loss. Moreover, IA feasibility constraint hinders its application to all users. Clustered IA provides a mechanism to mitigate the IA feasibility constraint and maximize the achievable rate. However, how to form IA clusters properly has not been well addressed. In this paper, we consider the problem of IA clustering in a multiuser interference channel at finite SNR. By exploiting the statistics of the channel state information (CSI), two novel clustering algorithms based on graph partitioning and heuristic are proposed. It is shown that the proposed schemes lead to significant gain in achievable rate when compared with non-cooperative transmission scheme.**

I. Introduction

The broadcast nature of wireless medium raises the issue of interference on multiuser communication. Traditionally, interference is avoided by allowing users to access the channel orthogonally, either in time or frequency domain. Recently, interference alignment (IA) provides a new approach to address this issue [1]. It designs all users' precoders and equalizers jointly such that the interference is aligned on the subspace of each receiver, whilst the desired signal can be transmitted on the interference-free space. Here the concept of space may refer to the extended symbol space in time/frequency domain or spatial space when multiple-input-multiple-output (MIMO) technique is applied.

Conventional IA schemes [1] [2] treated all interference signals with equal importance. In other words, their designs for precoders and equalizers only focused on aligning and canceling all the interference, but ignored the strength of the received signals. These designs are optimal in terms of achieving the optimal capacity scaling when the transmit power levels for all users go to infinity since, at asymptotically high signal-to-noise ratio (SNR), interference always dominates over noise. However, in practical communication systems with finite SNR, the achievable rate is affected mainly by the strong interference and is quite insensitive to the weak interference. Hence, eliminating the very weak interference by IA hardly improves the achievable rate. What is worse is that including these weak interferers in IA makes the performance worse due to the reduction of the signal space dimension. Therefore, it is not always optimal to employ IA to cancel all the interference. Moreover, IA may not always be feasible due to the IA feasibility constraint [3], especially with large number of users. Therefore, clustered IA has been proposed as a reasonable method to address the IA feasibility constraint issue while providing additional flexibility to optimize the achievable rate.

The idea of clustering has been introduced in wireless communications, e.g., the channel allocation problem is addressed in [4] via user clustering. The application of clustering for IA was considered in a cellular network and *ad hoc* network in [5] and [6], respectively, where IA was applied independently to small user groups referred to as IA clusters. Performance gains in achievable rate [5] and outage probability [6] by employing clustered IA have been reported. However, not much work has studied the issue on how to form the IA clusters. For example, [5] just studied the single cluster scenario and it is not clear how to form multiple clusters. Moreover, the single cluster in [5] was formed in a *predetermined* fashion, i.e., simply grouping the neighboring cells without further exploiting the channel state information (CSI).

In this paper, we address the issue of IA clustering in a K-user interference channel. Two clustering algorithms based on graph partitioning and heuristic are proposed. The idea is to form IA clusters intelligently by exploiting the statistics of the CSI in such a way that the relatively strong interference is captured among *intra-cluster* interference and is effectively suppressed by IA. The reason why we use the term "suppressed" instead of "canceled" is mainly due to the fact that in order to achieve a higher data rate, the precoder and equalizer designs should not only focus on canceling the intra-cluster interference but should also take into consideration the *inter-cluster* interference. To incorporate the effect of the inter-cluster interference, we model them by additive white noise. This requirement becomes more critical especially when strong inter-cluster interference exits, which dramatically increases the equivalent noise level. Here we slightly modify the *Max-SINR* algorithm in [7] to make it applicable for the precoder and equalizer designs in our scenario. Simulation results show that significant gain in achievable rate can be achieved by the proposed IA clustering algorithms.

The rest of this paper is organized as follows. Section II introduces the system model. The graph-partition-based clustering algorithm and heuristic clustering algorithm are proposed in Section III and Section IV, respectively. Section

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

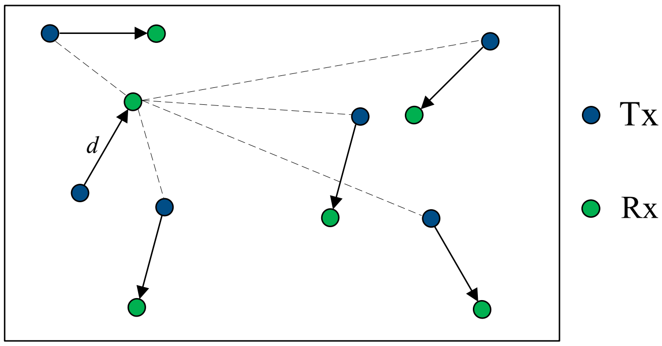

Fig. 1. K-user interference channel. The interference signals are indicated by the dashed lines.

V discusses the issue of strong inter-cluster interference and introduces the diagonal loading method to handle it. Simulation results are provided in Section VI. Finally, Section VII concludes the paper.

Notations: Bold upper case and lower case letters denote matrices and vectors respectively. $(\cdot)^T$ and $(\cdot)^H$ stand for transpose and Hermitian transpose, respectively. $|\cdot|$ denotes either the cardinality of a set or the absolute value of a scalar according to the scenario specified. $\preceq$ denotes componentwise inequality for vectors.

II. System Model

A. K-user Interference Channel

Consider a K-user interference channel (Fig. 1) where each transmitter has M antennas and each receiver has N antennas. Assume the ith transmitter sends one data stream s_i to the ith receiver, $i \in \{1, \dots, K\}$, with the transmitted signal $\mathbf{x}_i \in \mathbb{C}^{M\times 1}$ generated through a precoder $\mathbf{v}_i \in \mathbb{C}^{M\times 1}$, i.e., $\mathbf{x}_i = \mathbf{v}_i s_i$. The transmission power is given by $\mathbb{E}[\mathbf{x}_i^H \mathbf{x}_i] = P_i$. The output at the jth receiver can be expressed as

$$\mathbf{y}_j = \sum_{i=1}^{K} \sqrt{\rho_{ji}} \mathbf{H}_{ji} \mathbf{x}_i + \mathbf{z}_j, \tag{1}$$

where $\rho_{ji} = d_{ji}^{-\alpha}$ is the long term channel gain from the ith transmitter to the jth receiver which are separated by a distance d_{ji} (km), and α is the path-loss exponent. $\mathbf{H}_{ji} \in \mathbb{C}^{N\times M}$ is the normalized MIMO small scale Rayleigh fading channel gains from the ith transmitter to the jth receiver. $\mathbf{z}_j \in \mathbb{C}^{N\times 1}$ denotes the received noise which follows circularly symmetric Gaussian distribution with zero mean and identity covariance matrix, i.e., $\mathbf{z}_j \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$.

B. Clustered IA with IA Clustering

We need to make clear the concept of IA clustering first. We refer to the pair formed by a transmitter and its intended receiver as a transmitter-receiver (Tx-Rx) pair. Denote by $\mathcal{P} = \{1, 2, \dots, K\}$ the set of all Tx-Rx pairs. Each IA clustering is a partition on $\mathcal{P}$. Let $\mathbb{A}$ denote the collections of all possible clusterings for the K-user interference channel. Then, for each clustering $\mathcal{A} \in \mathbb{A}$, denote $\mathcal{A} = \{\mathcal{A}(1), \dots, \mathcal{A}(N^{\mathcal{A}})\}$, where $\mathcal{A}(m)$ is the mth cluster, $N^{\mathcal{A}}$ is the number of clusters and

$$\mathcal{A}(m) \subseteq \mathcal{P}, \quad \forall m, \tag{2}$$

$$\mathcal{A}(m) \cap \mathcal{A}(n) = \emptyset, \quad m \neq n, \tag{3}$$

$$\bigcup_{m=1}^{N^{\mathcal{A}}} \mathcal{A}(m) = \mathcal{P}. \tag{4}$$

Note that for fixed number of antennas M and N, the maximum number of Tx-Rx pairs L allowed in each cluster is restricted by the IA feasibility constraint [3]: $L \leq M + N - 1$. Therefore,

$$|\mathcal{A}(m)| \leq L, \quad \forall m. \tag{5}$$

Suppose the jth Tx-Rx pair belongs to the mth cluster under the clustering $\mathcal{A}$, i.e., $j \in \mathcal{A}(m)$. Its received signal can be expressed as

$$\begin{aligned} \mathbf{y}_j =& \sqrt{\rho_{jj}} \mathbf{H}_{jj} \mathbf{v}_j s_j + \sum_{i\neq j, i \in \mathcal{A}(m)} \sqrt{\rho_{ji}} \mathbf{H}_{ji} \mathbf{v}_i s_i \\ &+ \sum_{k \notin \mathcal{A}(m)} \sqrt{\rho_{jk}} \mathbf{H}_{jk} \mathbf{v}_k s_k + \mathbf{z}_j. \end{aligned} \tag{6}$$

According to clustered IA, the Tx-Rx pairs in the same IA cluster design the precoders and equalizers jointly in order to suppress the intra-cluster interference, while the inter-cluster interference is treated as noise. Therefore, at the jth receiver, only the interference coming from the transmitters that also belongs to $\mathcal{A}(m)$ will be aligned and canceled (if perfect IA within $\mathcal{A}(m)$ is applicable). That is,

$$\mathbf{u}_j^H \mathbf{H}_{ji} \mathbf{v}_i = 0, \quad i, j \in \mathcal{A}(m), i \neq j \tag{7}$$

$$\mathbf{u}_j^H \mathbf{H}_{jj} \mathbf{v}_j \neq 0, \tag{8}$$

where $\mathbf{u}_j \in \mathbb{C}^{N\times 1}$ is the equalizer at the jth receiver. Applying the equalizer $\mathbf{u}_j$ on $\mathbf{y}_j$, we obtain

$$\begin{aligned} y_j &= \mathbf{u}_j^H \mathbf{y}_j \\ &= \sqrt{\rho_{jj}} \mathbf{u}_j^H \mathbf{H}_{jj} \mathbf{v}_j s_j + \sum_{k \notin \mathcal{A}(m)} \sqrt{\rho_{jk}} \mathbf{u}_j^H \mathbf{H}_{jk} \mathbf{v}_k s_k + \mathbf{u}_j^H \mathbf{z}_j. \end{aligned} \tag{9}$$

The achievable rate for the jth Tx-Rx pair is given by

$$R_j = \log_2 \left(1 + \frac{\rho_{jj} P_j |\mathbf{u}_j^H \mathbf{H}_{jj} \mathbf{v}_j|^2}{1 + \sum_{k \notin \mathcal{A}(m)} \rho_{jk} P_k |\mathbf{u}_j^H \mathbf{H}_{jk} \mathbf{v}_k|^2} \right). \tag{10}$$

From (10), if we only consider the large scale path-loss effect of the channel, it is preferable that by applying the partition strategy $\mathcal{A}$, the residual interference $\sum_{k \notin \mathcal{A}(m)} \rho_{jk} P_k$ originated from the Tx-Rx pairs belonging to other clusters should be kept as small as possible such that R_j does not degrade too much. In other words, we want to design efficient clustering algorithms subject to the IA feasibility constraint such that the inter-cluster interference is minimized.

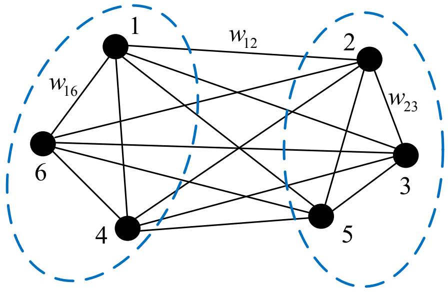

Fig. 2. A K-user interference channel can be interpreted as a complete graph. Each vertex represents a Tx-Rx pair. The ellipses show an example of the clustering.

III. Graph-Partition-based Clustering Algorithm

A. Interpreting a K-user Interference Channel as a Graph

In graph theory, a graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a finite set of vertices and $\mathcal{E}$ is a finite set of edges. We can interpret a fully connected K-user interference channel as a *weighted* graph (Fig. 2). The vertex set is given by $\mathcal{P}$ and the weight of each edge joining two vertices represents the cost of a relationship between these two Tx-Rx pairs. Let $\mathcal{W} = \{w_{ij} : i, j \in \{1, \ldots, K\}, i \leq j\}$ be the set of weights and the element w_{ij} is related to the statistics of CSI of the interference channel through a mapping:[1]

$$w_{ij} = \frac{\rho_{ij} P_j}{\rho_{ii} P_i} + \frac{\rho_{ji} P_i}{\rho_{jj} P_j}, \quad \forall i, j,\ i \leq j. \tag{11}$$

It roughly measures the inter-pair interference normalized by the strength of intended signals between two Tx-Rx pairs. Hence, large weight implies strong interference between the pairs and they tend to be included in the same cluster. Similar usage of the interference measurement (11) was used in [8].

B. Graph Partitioning Formulation for IA Clustering

The problem of IA clustering can be formulated as a graph partitioning problem. Our objective is to minimize the aggregated inter-cluster interference, or equivalently, maximize the aggregated interference for all clusters. With (11), this requirement can be translated to the objective that the sum of the edge weights within all clusters is maximized, provided that each cluster has no more than L vertices (Tx-Rx pairs). Therefore, we have the following optimization problem:

$$\begin{aligned} \underset{\mathcal{A} \in \mathbb{A}}{\text{maximize}} \quad & \sum_{m=1}^{N^{\mathcal{A}}} \left(\sum_{i,j \in \mathcal{A}(m)} w_{ij} \right) \\ \text{subject to} \quad & (2), (3), (4) \text{ and } (5). \end{aligned} \tag{12}$$

Graph partitioning has been known as an NP hard problem [9], thus generally it is not efficient to solve the problem (12) directly by evaluating all possible partitions. Hence we resort to suboptimal but more efficient algorithms.

[1] The ambiguous "weight" w_{jj} can be treated as the weight of the vertex. Moreover, as to the weight assignment (11), $w_{jj} = \text{const}, \forall j$. Thus it only introduces a constant to all vertices and does not affect the partitioning result.

C. Low Complexity Algorithm through Set Partitioning

We introduce a different formulation of graph partitioning which uses the concept of set partitioning. This formulation was used in [9] [10] to design suboptimal but efficient graph partitioning algorithms.

Let $\mathcal{A}(k) \subseteq \mathcal{P}$ be a cluster of the graph $\mathcal{G}$. Let x_k be a *binary* variable that will be 1 if $\mathcal{A}(k)$ is used in the solution to the clustering problem and 0 otherwise. Let $a_k = \sum_{i,j \in \mathcal{A}(k)} w_{ij}$ be the sum weights of cluster $\mathcal{A}(k)$. For each vertex $i \in \mathcal{P}$, $y_i^k = 1$ if $i \in \mathcal{A}(k)$, otherwise $y_i^k = 0$. Then the clustering problem can be formulated as

$$\begin{aligned} \text{maximize} \quad & \sum_k a_k x_k \\ \text{subject to} \quad & \sum_k y_i^k x_k = 1, \forall i && (13) \\ & \sum_i y_i^k \leq L, \forall k && (14) \\ & x_k \in \{0, 1\}, \forall k, && (15) \end{aligned}$$

where (13) means each vertex only belongs to one cluster, and (14) is the constraint for the cluster size.

If we relax the binary constraint (15) to $0 \leq x_k \leq 1$, the relaxed problem becomes a linear programming (LP) problem. We can express the LP relaxation problem in a matrix form

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{a}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{Y}\mathbf{x} = \mathbf{1}_{K \times 1} \\ & \mathbf{0}_{N_A \times 1} \preceq \mathbf{x} \preceq \mathbf{1}_{N_A \times 1}, \end{aligned} \tag{16}$$

where $N_A = \sum_{i=1}^{L} \binom{K}{i}$ is the number of feasible clusters which satisfy the cluster size constraint (14). $\mathbf{x} = [x_1, \ldots, x_{N_A}]^T$, $\mathbf{a} = [a_1, \ldots, a_{N_A}]^T$ and $\mathbf{Y} = \begin{bmatrix} y_1^1 & \cdots & y_1^{N_A} \\ \vdots & \ddots & \vdots \\ y_K^1 & \cdots & y_K^{N_A} \end{bmatrix}$. The relaxed problem (16) can be solved efficiently by LP solvers. If the optimal solution $\mathbf{x}^*$ of (16) only has binary elements, i.e., $x_i^* \in \{0, 1\}, \forall i$, we automatically derive the optimal solution for the original clustering problem. However, $\mathbf{x}^*$ does not always attain this property. Hence, further processing is required. Note that we cannot simply round the elements of $\mathbf{x}^*$ to nearest integers because it may give a result which violates the constraint (13). Intuitively, we can think of those clusters which correspond to $x_i^* > 0$ as candidate clusters. Then we search for the *non-intersecting* clusters among these candidates which result in the maximum sum weights. The search is not so complicated because most of x_i^* will be zero thus the searching space can be greatly reduced compared to (12). Finally, if there exists some Tx-Rx pairs that have not been included in any cluster, we just group them randomly to form new clusters. Algorithm 1 summarizes the graph-partition-based clustering algorithm.

Except dealing with the fractional solution $\mathbf{x}^*$. Another issue in solving this LP relaxation problem is that the number of feasible clusters N_A becomes very large as K grows. Hence strictly speaking it is still not efficient to solve (16) directly in certain scenarios. More carefully designed graph partitioning

Algorithm 1 Graph-partition based clustering algorithm

Solve the LP relaxation problem (16) and obtain $\mathbf{x}^*$.
if $x_i^* \in \{0,1\}, \forall i$ **then**
  Output the clusters indicated by $x_i^* = 1$ as the clustering result.
else
  Form a candidate set $\mathcal{S} = \{\mathcal{A}(i) : x_i^* > 0\}$.
  For all possible $\mathcal{S}_{\mathcal{A}} = \left\{\mathcal{A}(j) : \mathcal{A}(j) \in \mathcal{S} \text{ and } \bigcap_j \mathcal{A}(j) = \emptyset\right\} \subseteq \mathcal{S}$, choose the optimal $\mathcal{S}_{\mathcal{A}}^*$ such that $\sum_{\mathcal{A}(j)\in\mathcal{S}_{\mathcal{A}}} a_j$ is maximized.
  Output the clusters $\mathcal{A}(j) \in \mathcal{S}_{\mathcal{A}}^*$ as the resulting clusters.
  if $\mathcal{P} \setminus \bigcup_{\mathcal{A}(j)\in\mathcal{S}_{\mathcal{A}}^*} \mathcal{A}(j) \neq \emptyset$ **then**
    Randomly form and output new clusters for the remaining pairs.
  end if
end if

algorithms based on cutting-plane method or column generation can be found in [9] [10] and the references therein.

IV. Heuristic Clustering Algorithm

In this section, we propose a low complexity heuristic algorithm to form the IA clusters. Because of large variations on the strength of received signals at each receiver, we assign each Tx-Rx pair a priority index such that the pair which is interfered most has the highest priority. During the process of cluster forming, the pairs with higher priorities will be addressed first because they see relatively strong interference, and eliminating the interference for these pairs usually results in significant improvement on the achievable rate.

Algorithm 2 summarizes the proposed clustering algorithm. Note that step 6 checks the aggregated interference caused by the remaining pairs that have not been included in any cluster. If they only contribute very weak interference, then we stop adding Tx-Rx pairs to the current cluster.

Compared with Algorithm 1, heuristic algorithm works more like a greedy algorithm as it tends to benefit the Tx-Rx pair which sees stronger interference (with higher priority).

V. Precoder and Equalizer Designs for IA

In this section, we briefly discuss the precoder and equalizer designs for clustered IA. Since the focus is on achieving higher data rates, the *Max-SINR* algorithm [7] is used. The details of the algorithm is ignored here. However, we want to point out that the original design of this algorithm did not address the uncoordinated interference, hence it may suffer performance degradation when strong inter-cluster interference exists (see the rate decrease in Fig. 3). A simple method to address this issue is that we adaptively increase the white noise level in the covariance matrix of interference plus noise (equation (30) of [7]):

$$\mathbf{C}^{[j]} = \sum_{i\neq j,\ i\in\mathcal{A}(m)} \rho_{ji} P_i \mathbf{H}_{ji}\mathbf{v}_i\mathbf{v}_i^H\mathbf{H}_{ji}^H + (1+\gamma_j)\mathbf{I}_N, \quad (17)$$

where $\gamma_j = \sum_{k\notin\mathcal{A}(m)} \rho_{jk} P_k$ represents the increase of the noise level by treating the inter-cluster interference as noise.

Algorithm 2 Heuristic clustering algorithm

1: Compute the aggregated interference $I_j = \sum_{i\neq j} \rho_{ji} P_i, \forall j$, and sort the pairs based on I_j in descending order.
2: Start from $m = 1$.
3: Set $\mathcal{A}(m) = \emptyset$.
4: Add the j^*th pair to $\mathcal{A}(m)$, where $j^* = \underset{j\in\mathcal{P}\setminus\bigcup_{i=1}^m \mathcal{A}(i)}{\operatorname{argmax}} I_j$.
5: **while** $\mathcal{P}\setminus\bigcup_{i=1}^m \mathcal{A}(i) \neq \emptyset$ **do**
6:   **if** $|\mathcal{A}(m)| = L$ or $\sum_{k\in\mathcal{P}\setminus\bigcup_{i=1}^m \mathcal{A}(i)} \rho_{jk} P_k < \min\{1, \rho_{jj} P_j\}, \forall j \in \mathcal{A}(m)$ **then**
7:     Jump to step 12.
8:   **else**
9:     Add the k^*th pair to $\mathcal{A}(m)$, where $k^* = \underset{k\in\mathcal{P}\setminus\bigcup_{i=1}^m \mathcal{A}(i)}{\operatorname{argmax}} \left(\sum_{i\neq j,\ i,j\in\mathcal{A}(m)\cup\{k\}} \rho_{ji} P_i \right)$.
10:   **end if**
11: **end while**
12: **if** $\mathcal{P}\setminus\bigcup_{i=1}^m \mathcal{A}(i) \neq \emptyset$ **then**
13:   Set $m = m + 1$ and jump to step 3.
14: **else**
15:   Output $\mathcal{A}(1), \mathcal{A}(2), \cdots, \mathcal{A}(m)$ as the clustering results.
16: **end if**

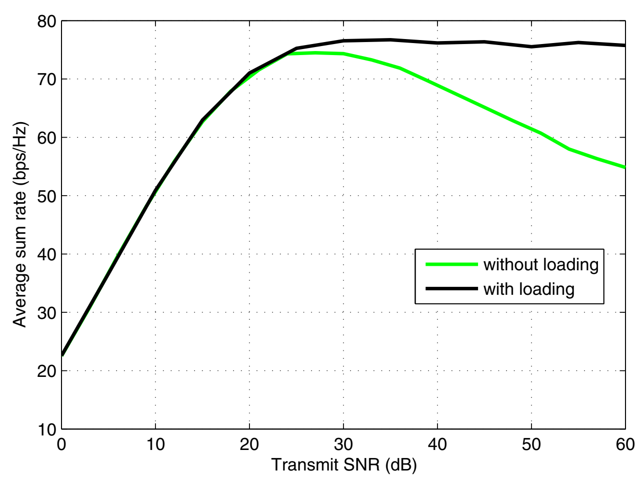

Fig. 3. The achievable rates with/without diagonal loading for 12-user interference channel with $M = N = 3$ and $L = 3$. All users are transmitting at the same transmit SNR.

This method is referred to as diagonal loading and it does not require instantaneous CSI from other clusters. For $\gamma_j \gg 1$, strong inter-cluster interference will be taken into consideration to make sure it does not degrade the achievable rate (Fig. 3).

VI. Simulation Results

Simulation results are provided in this section to evaluate the performance of proposed algorithms. Assume all the transmitters are uniformly distributed in a square area (10km ×10km), the distance between each transmitter and its intended receiver is given by $d_{jj} = 1$km, $\forall j$. The path-loss exponent $\alpha = 3.76$. The performances of four schemes are considered:

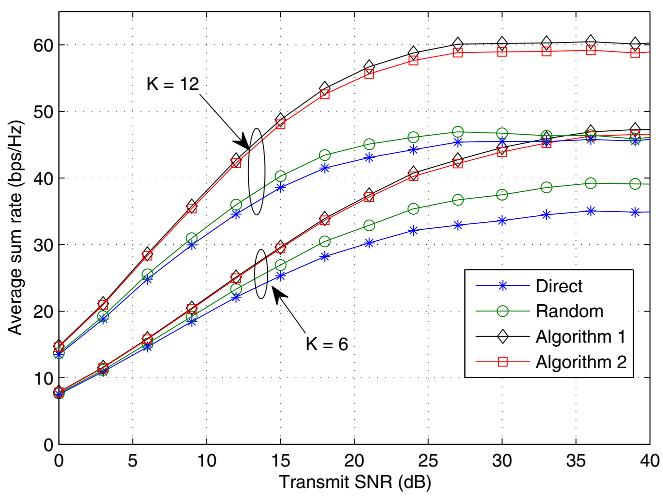

Fig. 4. Sum rate versus transmit SNR for $M = N = 2$ and $L = 3$.

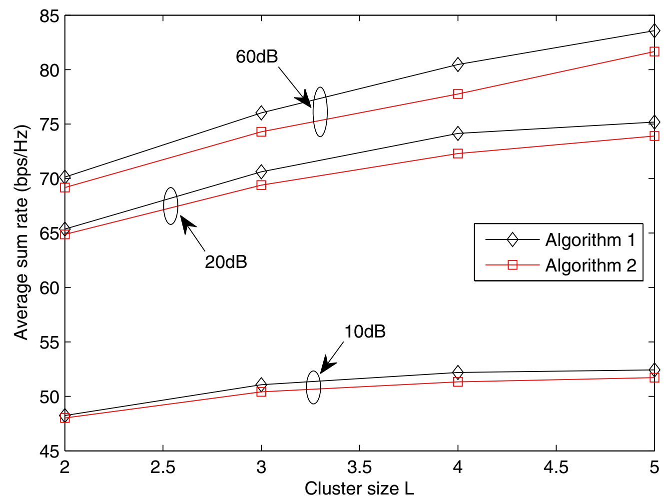

Fig. 5. Sum rate versus L for $M = N = 3$ and $K = 12$.

- Direct transmission without any cooperation.
  In this scheme, each pair transmit/receive its data stream along the direction corresponding to the largest singular value of the channel. That is, $\mathbf{u}_j^H \mathbf{H}_{jj} \mathbf{v}_j = \max \lambda(\mathbf{H}_{jj}), \forall j$. This scheme serves as the baseline.
- IA with random clustering.
  In this scheme, we randomly choose Tx-Rx pairs to form IA clusters provided that the cluster size is less than or equal to L.
- IA using Algorithm 1 in forming IA clusters.
- IA using Algorithm 2 in forming IA clusters.

Fig. 4 shows the average sum rate versus the transmit SNR for 6-user and 12-user interference channel, respectively. The achievable rate increases as the increase of transmit SNR, but eventually it saturates because of irreducible interference. Note that for less populated system ($K = 6$), the transmit SNR at which the sum rate saturates is larger than that of a denser system ($K = 12$) because it has more budget to accommodate interference. Both clustering algorithms provide significant gains over the direct transmission scheme. Moreover, we can see that random clustering only provides marginal gain in term of sum rate. This result states the importance of forming proper IA clusters and we could only enjoy the benefit of clustered IA if the CSI can be exploited.

Fig. 5 shows how the cluster size affects the sum rate at different transmit SNR. Note that the larger the cluster size, the higher sum rate can be achieved, provided that IA is feasible. This is due to more intra-cluster interference can be suppressed by IA as well as the reduction of inter-cluster interference. Moreover, the rate improvement is more significant at high SNR, since suppressing interference becomes more critical at interference-limited regime. The issue of using larger cluster size is that it leads to higher signaling overhead within the cluster when designing the precoders and equalizers for IA.

VII. Conclusion

Clustered IA has been proposed as a method to mitigate the IA feasibility constraint and maximize the achievable rate for multiuser communication. In this paper, we address the issue of IA clustering in a multiuser interference channel at finite SNR. Noting that signals received at a receiver usually have distinct strengths because of path-loss effect, we propose two clustering algorithms by exploiting the statistics of the CSI. IA clusters are formed such that strong interference is captures among intra-cluster interference while inter-cluster interference is relatively weak. Therefore, clustered IA effectively suppresses the intra-cluster interference and leads to significant gains on the achievable rate.

References

[1] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[2] ——, "Interference alignment and the degrees of freedom of wireless X networks," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3893–3980, Sep. 2009.

[3] C. M. Yetis, T. Gou, S. A. Jafar, and A. H. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Trans. Signal Process.*, vol. 58, pp. 4771–4782, Sep. 2010.

[4] Y. Kasdani, Y. H. Chew, C. Yuen, and W. H. Chin, "Channel allocation in a multiple distributed vehicular users using game theory," in *IEEE VTC 2010 Spring*, Taipei.

[5] R. Tresch, M. Guillaud, and E. Riegler, "A clustered alignment-based interference management approach for large OFDMA cellular networks," in *Joint NEWCOM++/COST2100 Workshop on Radio Resource Allocation for LTE*, 2009.

[6] R. Tresch and M. Guillaud, "Performance of interference alignment in clustered wireless ad hoc networks," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Austin, USA, 2010.

[7] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, pp. 3309–3322, Jun. 2011.

[8] H. Huang and V. K. N. Lau, "Partial interference alignment for K-user MIMO interference channels," *IEEE Trans. Signal Process.*, vol. 59, pp. 4900–4908, Oct. 2011.

[9] X. Ji and J. Mitchell, "Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement," *Discrete Optimization*, vol. 4, pp. 87–102, 2007.

[10] E. Johnson, A. Mehrotra, and G. Nemhauser, "Min-cut clustering," *Mathematical programming*, vol. 62, no. 1, pp. 133–151, 1993.