Distributed Joint AP Grouping and User Association for MU-MIMO Networks

Wangkit Wong S.-H. Gary Chan
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Hong Kong, China
Email: {wwongaa, gchan}@cse.ust.hk

***Abstract*—Interference Alignment (IA) has emerged as a promising interference coordination approach for cooperative MIMO systems. Due to heavy CSI feedback overhead, APs (Access Points) need to be partitioned into cooperation groups no larger than a certain size where only APs in the same group are able to cooperate with IA. We consider a general MIMO network using a hybrid interference coordination approach, i.e. intra-group interference is managed with IA, while inter-group interference is overcome with traditional orthogonal multiple access techniques. Users are usually non-uniformly distributed. Their throughput can be improved by association optimization.**

We study the novel problem of minimizing AP load by joint AP grouping and user association. The problem is shown to be NP-hard. Based on alternating direction optimization, we propose DAGA (Distributed Joint AP Grouping and User Association) to tackle the problem. DAGA is distributed and uses only long-term CSI. Based on current AP grouping, it produces an approximated user association solution which is at most $e \log m$ (m is the number of APs) times of the optimum. Based on current user association, it adjusts AP grouping with local search. Extensive simulation results show that it substantially outperforms other comparison schemes.

***Index Terms*—MU-MIMO network, Interference alignment, Load balancing, Joint Optimization, Approximation algorithm**

I. Introduction

To meet the escalating traffic demand, APs (Access Points) are being densely deployed. While AP densification is able to improve received signal strength, it is accompanied with strong co-channel interference. Classical interference coordination approaches orthogonalize spectral access in both frequency and time domains, which might lead to low spectral efficiency. Recently, a promising interference coordination scheme termed Interference Alignment (IA) has been proposed. In IA, APs cooperate with each other. Each cooperating AP transmit signal with zero-forcing precoding to its users such that the interference is aligned to a null space of each user [1], [2]. In such way, interference alignment nullifies interference at users covered by multiple APs.

In the ideal case, all APs cooperate together. However, IA requires CSI (Channel State Information) of users. To obtain CSI, APs in the same cooperation group perform channel sounding/training together [2]. During the process, users feed back CSI to all APs cooperating with its home AP. Based on the CSI, APs in the cooperation group perform user scheduling, then transmit to selected users independently. In realistic systems, the number of cooperating APs is limited by coordination signaling. Therefore, APs should be partitioned into cooperation groups of proper size. Intra-group interference is hence managed via IA, while inter-group interference is managed using traditional time division/frequency division interference coordinated approaches. We show an example in Figure 1. AP A and AP B form a cooperation group. User 2 and user 3 are scheduled. AP A nullifies its interference at user 3 while transmitting to user 2. AP C outside the cooperation group cannot perform interference alignment for user 3 due to the absence of CSI. Hence, it has to transmit in different time slots.

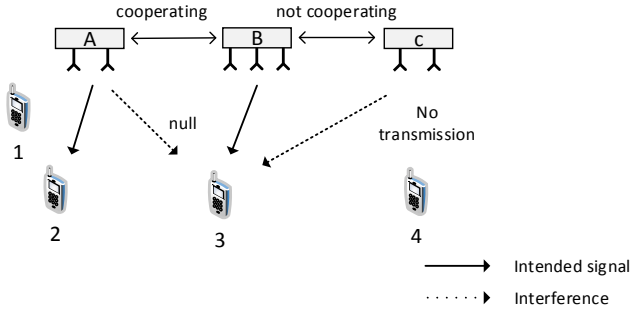

Fig. 1. Interference alignment between cooperative APs.

AP Grouping, i.e., how to partition APs into cooperation groups, is an important factor in network performance because different grouping solutions lead to different interference patterns. Previous art optimizes AP grouping jointly with beamforming on a per-packet basis in adapting to user channel dynamics, which incurs a large amount of overhead in terms of global user scheduling and full instantaneous CSI feedback from users. To address this, we perform AP grouping in a larger time scale than beamforming using only long-term CSI.

AP coverages may overlap and interfere with each other. *User association* is to assign users to one of the APs for service. The traditional approach selfishly picks the AP with the strongest signal. Since user traffic is usually non-uniform, AP load might be unbalanced with clusters of users associated with some "hot" APs while some "colder" ones with few users.

This work was supported, in part, by The Hong Kong R&D Center for Logistics and Supply Chain Management Enabling Technologies (ITP/046/15LP and ITT/002/17LP) and Guangzhou Science and Technology Innovation Commission (GZSTI16EG14).

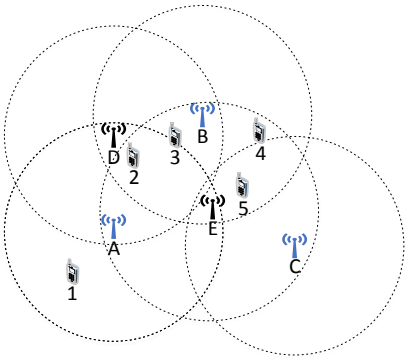

Fig. 2. An illustration of the need for joint optmization.

To balance AP load, previous works propose online user association algorithms, which select an AP for each joining user. The user will not change its association once associated with an AP. However, the network is dynamic with user departure. As a result, the performance tends to degrade over time. We hence need to re-optimize association continuously by migrating some users to lightly loaded APs.

Different decisions of user association form different communication links, and hence affect AP grouping. On the other hand, AP grouping decisions determine the beamforming/IA capability of APs, and hence affect user association decisions. Therefore, user association and AP grouping are coupled and ask for joint consideration.

We show an illustrative example in Figure 2. The coverage of each AP is represented by a disk centered at the AP. AP A, B and C operate on channel 1, while AP D and E operate on channel 2. Whether AP B cooperates with AP C depends on the association decision of user 5. If user 5 associates with AP B or C, AP B and C should cooperate to mitigate the interference at user 5. However, if user 5 associates with AP E, AP B may cooperate with AP A instead of C because there are no users in the overlapping coverage of AP B and C. On the other hand, the association decision of user 2 depends on grouping decisions. If AP A does not cooperate with B, user 2 should associate with AP D as AP D is the closest AP. However, user 2 receives less interference by associating with AP B in the case where AP A cooperates with B.

In this work, we consider general MU-MIMO wireless networks, such as WLANs or small cell networks. Each AP has a certain, possibly heterogeneous, number of antennas, with some pre-assigned frequency channels. Each user has a certain traffic demand. Channel utilization is an important indicator of network performance [3]. We study the joint optimization for AP grouping and user association to minimize the heaviest AP load (which is defined as channel utilization at the AP). To reduce signaling overhead, we consider the system operates over two time scales. AP grouping and user association are optimized using long-term CSI over long-term time scale. User scheduling and beamforming design are carried out over short-term time scale.

Our problem is novel and merits a new study because previous work has not considered the joint optimization of AP cooperation group forming and user association. We approach the problem via the following:

1) *Problem formulation:* We formulate the joint optimization problem for AP grouping and user association. We show that it is NP-hard.

2) *Distributed low-complexity algorithm:* We propose DAGA (**D**istributed Joint **A**P **G**rouping and User **A**ssociation) to tackle the problem. DAGA is a fully distributed algorithm based on alternating direction optimization. It alternates between association optimization and AP grouping optimization. Given AP grouping, DAGA optimizes user association in an online manner. DAGA produces at most $e \log m$ (where m is the number of APs) of the optimum AP load. It is worth noting that previous online approximation algorithms [4], [5] can only provide performance guarantee when there is no user departure. DAGA, on the contrary, adopts association re-optimization to contain user departure and maintains the performance guarantee. Based on current user association, DAGA optimizes AP grouping using low-cost local search. Moreover, DAGA only uses long-term CSI and hence it is practical and robust.

3) *Extensive simulations studies:* We demonstrate the effectiveness of DAGA using MAC-level simulations. Simulation results show that DAGA outperforms recent schemes by a wide margin in terms of throughput and fairness.

The rest of the paper is organized as follows. We first discuss related work in Section II, followed by presenting the system models and problem formulation in Section III. In Section IV we present CA (Cooperation-aware Association), an efficient approximation algorithm for user association given current AP grouping. Inspired by CA, we present DAGA for joint AP grouping and user association in Section V. We present in Section VI illustrative simulation studies. We conclude in Section VII. Important proofs are provided in the Appendix.

II. Related works

Previous works on user association can be broadly categorized into user association for single antenna networks and user association for MIMO networks. Many works study on-line user association [5]–[8]. Most of these schemes assign an AP to a user when it is joining the network. Although simple, it is unclear how they perform compared with the optimum. Some works [4], [9] provide approximation/competitive bounds. However, the performance bounds only hold when there is no user departure. In contrast, DAGA provides performance guarantee even if there are user departures by locally re-optimizing user association. There have been works [10], [11] optimizing user association based on the assumption of no inter-AP interference. In reality, APs may interfere with each other. It is hence important to consider user association in the presence of such interference. Our optimization well captures interference and coordinate it with IA and orthogonal channel access.

There have been works studying user association in MIMO networks. Some of these schemes [12]–[14] optimize user association using instantaneous CSI. These works jointly design cell association and beamformer to optimize some utility, such as weighted sum rate or energy consumption. They incur high overhead as optimal cell association changes per coherent time. To reduce overhead, user association schemes based on long-term channel condition have been proposed [3], [15], [16]. These schemes optimize ergodic user performance using CSI statistics instead of instantaneous CSI. All of these schemes optimize user association without the awareness of AP cooperation.

Another line of research studies AP grouping. There have been works studying static AP grouping [2], [17], [18]. Such approaches group nearby APs into clusters and hence is not adaptive to user distribution. Much works have study per-slot dynamic AP/user clustering using instantaneous CSI [19], [20], which incur high signaling overhead. Recent works address overhead by clustering APs using only long-term CSI [21], [22].

III. System Model and Problem Formulation

A. Network Model

We consider a general MU-MIMO wireless network, such as a WiFi network or a small cell network. The network is formed by a number of Access Points (AP) and users. Let $\mathbf{A}$ be the set of APs and m be its cardinality. Denote the set of users as $\mathbf{U}$, where $|\mathbf{U}| = n$. Users in the network are dynamic (in the sense that they may join or leave at any time). AP i has K_i transmit antennas (K_i is also referred as the DoF of the AP). APs may operate on different frequency channels. The channel AP i uses is denoted as F_i. The total transmit power budget of AP i is P_i. User j has K_j antennas.

The wireless channel from AP i to user j is denoted as $H_{ji} \in \mathbf{C}^{K_j \times K_i}$. Each element of H_{ji} is composed by distance based path-loss and fast fading channel coefficient. That is $[H_{ji}]_{pq} = [h_{ji}]_{pq}\gamma_{ji}$, where γ_{ji} is the long-term channel gain, determined by the distance and shadow fading between user j and AP i. Complex number $[h_{ji}]_{pq}$ is the fast fading channel coefficient between the pth receive antenna and the qth transmit antenna. The set of users associated with AP i is denoted as $\mathbf{U}_{asso}(i)$.

The network is represented by a directed graph $G(\mathbf{V}, \mathbf{L})$, where $\mathbf{V} = \mathbf{A} \cup \mathbf{U}$ is the set of network nodes and $\mathbf{L}$ is the set of potential communication links. We say node j is within the interference range of node n if the reception of node j is interfered by the transmission of node n. That is the received power at j from n is above some signal detection threshold (e.g. the signal detection threshold in 802.11 is -82 dBm for 20 MHz channels). We define $\mathbf{Cover}_{by}(n)$ as the set of nodes those are covered by the interference range of node n. Similarly, the set of nodes covering node j is denoted as $\mathbf{Cover}(j)$.

In MU-MIMO networks, an AP transmits multiple data streams to associated users simultaneously. Inter-stream interference is mitigated with MU-beamforming. Moreover, an AP can apply interference alignment to mitigate its interference at nodes in the coverage of its interference range. Both MU-beamforming and interference alignment are achieved by zero-forcing beamforming.

Consider that when user j is receiving a signal from AP i, APs in $\mathbf{Cover}(j)$ are transmitting simultaneously. AP i uses $V_{ji} \in \mathbb{C}^{K_i \times 1}$ as a linear precoder for user j. Similarly, user j uses $R_j \in \mathbb{C}^{1 \times K_j}$ as receive filter. Denote the symbol transmitted to user j as x_j. The instantaneous received signal at user j is given by

$$y_{ji} = \underbrace{R_j H_{ji} V_{ji} x_j}_{\text{desired signal}} + \underbrace{\sum_{k \in \mathbf{U}_{asso}(i), k \neq j} R_j H_{ji} V_{ki} x_k}_{\text{inter-stream interference}} + \underbrace{\sum_{l \in \mathbf{Cover}(j)} \sum_{m \in \mathbf{U}_{asso}(l), m \neq j} R_j H_{jl} V_{ml} x_m}_{\text{inter-AP interference}} + \underbrace{R_j z_j}_{\text{noise}}. \quad (1)$$

In order to eliminate inter-stream interference, precoding vector V_{ki} is designed such that $H_{ji}V_{ki} = 0, \forall j \neq k$. AP l aligns its signal according to V_{ml} to nullify its interference at user j ($H_{jl}V_{ml} = 0, \forall m \in \mathbf{U}_{asso}(l)$). Clearly, instantaneous signal $R_j H_{ji} V_{ji} x_j$ received at user j is hence affected by precoder V_{ji}.

Similar to previous work [21], we consider that V_{ji} does not depend on H_{ji}. Consider that each AP i allocates power P_i/K_i to each of its users. The effective desired signal received $R_j H_{ji} V_{ji} \sim \mathcal{CN}(0, \gamma_{ji} P_i / K_i)$. That is the distribution of received signal strength only depends on long-term channel γ_{ji} and the power allocated. Therefore, we optimize AP cooperation and user association based on long-term CSI. The expected link rate between AP i and user j is estimated as

$$C_{ij} = W \log\left(1 + \frac{P_i \gamma_{ij}}{K_i N_0 W}\right), \quad (2)$$

where W is the bandwidth and N_0 is the per-spectral noise density.

The MU-beamforming and interference alignment capability of an AP is determined by the number of antennas it has. An AP can nullify interference for I_n receive antennas while transmitting S_n data streams, if $I_n + S_n \leq K_n$, which is referred as DoF constraint [23], [24].

We consider that each user receives only one stream at a time. Nullifying interference for any user consumes 1 DoF no matter how many antennas a user has (because we only need to null interference at any one of the antennas). To perform interference alignment, joint channel sounding/training between cooperating APs is required. We hence consider APs are divided into multiple cooperation groups. We consider that different users may have different traffic demand, which is determined by the applications running on the user devices. Denote d_j as the traffic demand of user j.

B. Problem Formulation

The notations we use have been summarized in Table I. Denote the set of cooperation groups as $\mathbf{G}$. By allowing empty groups, we consider m APs are grouped into m cooperation

TABLE I
MAJOR SYMBOLS USED IN THIS PAPER.

Notation	Definition
F_i	frequency channel used by AP i
d_j	traffic demand of user j (bits/s)
r_{ij}	the expected link rate between AP i and user j (bits/s)
$\mathbf{Cover}_{by}(n)$	the set of nodes those are covered by the interference range of node n
$\mathbf{Cover}(j)$	the set of nodes whose interference ranges cover node j
$\mathbf{G}$	the set of cooperation group
z_{ki}	binary variable indicating whether AP i is in the cooperation group k
y_n^i	binary variable indicating whether AP i and n are in the same cooperation group
x_{ij}	binary variable indicating whether user j associates with AP i

groups, i.e. $|\mathbf{G}| = m$. Let binary variable z_{ki} indicate whether AP i is in cooperation group k. In order for the cooperation signaling overhead to be affordable, we require the size of each group to be at most S^{max}. The value of S^{max} is determined by the backhaul capability and channel coherent time. We hence have the following group size constraint.

$$\sum_{i\in\mathbf{A}} z_{ki} \leq S^{max}, \forall k \in \mathbf{G}. \tag{3}$$

Due to the fact that each AP can be assigned to only one group, we have

$$\sum_{k\in\mathbf{G}} z_{ki} = 1, \forall i \in \mathbf{A}. \tag{4}$$

We further use binary variable y_n^i to indicate whether AP n and AP i are grouped into the same group. Clearly, it can be calculated as the following: $y_n^i = \sum_{k\in\mathbf{G}} z_{kn} z_{ki}, \forall n, i \in \mathbf{A}$.

Let x_{ij} be the binary variable indicating whether user j associates with AP i. Each user must associate with only one AP for data access, i.e.

$$\sum_{i\in\mathbf{A}} x_{ij} = 1, \forall j \in \mathbf{U}. \tag{5}$$

Consider an AP i and a user in the interference range of i, i.e. $j \in \mathbf{Cover}_{by}(i)$. If user j associates with AP i, it utilizes the channel of i for $\frac{d_j}{C_{ij}}$ amount of time. Since AP i can transmit to $K_i - 1$ other users when serving j, user j consumes 1 DoF of i. The channel utilization at AP i due to user j is hence $\frac{d_j}{C_{ij}K_i}$.

If user j is not associated with AP i, it still occupies the channel of i if it associates with an AP n which uses the same channel as i. There are two cases. In the first case, AP i and AP n are in the same cooperation group. When AP n is sending to user j, i can transmit simultaneously by aligning its transmit signal to the null space of user j. This IA operation consumes 1 DoF of i. The channel utilization at AP i due to user j is hence $\frac{d_j}{C_{nj}K_i}$.

In the second case, AP i does not cooperate with AP n. When user j is active, AP i cannot access the channel otherwise it interferes user j. This means j consumes all the DoF of i. Therefore, the channel utilization at AP i due to j is $\frac{d_j}{C_{nj}}$. We define ρ_{nj}^i as the channel utilization at AP i due to user j given that j associates with AP n. Summarizing all cases,

$$\rho_{nj}^i = \begin{cases} 0, & \text{if } F_n \neq F_i \text{ or } j \notin \mathbf{Cover}_{by}(i)\ ; \\ \frac{d_j}{C_{ij}K_i}, & \text{if } n = i; \\ \frac{d_j}{C_{nj}K_i}, & \text{if } n \neq i,\ y_n^i = 1; \\ \frac{d_j}{C_{nj}}, & \text{if } n \neq i,\ y_n^i = 0. \end{cases} . \tag{6}$$

We define the aggregated channel utilization due to all users as the load of AP i, calculated as

$$L_i = \sum_{j\in\mathbf{U}} \sum_{n\in\mathbf{A}} x_{nj}\rho_{nj}^i. \tag{7}$$

For load balancing, we seek to minimize the maximum AP load across the whole network. Specifically, the joint AP grouping and user association problem can be formally express as

$$\min \max_{i\in\mathbf{A}} L_i, \tag{8}$$
$$\text{s.t. Constraints } (3),\ (4),\ (5)$$
$$z_{kj},\ x_{ij} \in \{0,1\}, \forall k \in \mathbf{G}, i \in \mathbf{A}, j \in \mathbf{U}$$

The problem is an integer non-linear non-convex optimization problem. We show in the next section the problem is NP-hard even in the case where AP grouping is given. Therefore, the joint optimization problem is NP-hard.

IV. CA: COOPERATION-AWARE USER ASSOCIATION

In this section, we consider the case where AP cooperation grouping is fixed. The joint problem then reduces to the following user association problem (UAP): $\min \max_{i\in\mathbf{A}} L_i$, subject to $\sum_{i\in\mathbf{A}} x_{ij} = 1, \forall j \in \mathbf{U}$. We first analyze the hardness of UAP. Then, we propose CA (**C**ooperation-aware **A**ssociation) to tackle the problem. CA is an approximation algorithm producing at most $e \log m$ times AP load compared with the optimal solution.

We consider that users may join and leave the network at any time. The PUMSP (Parallel Unrelated Machine Scheduling Problem) can be reduced to UAP with fixed user set. Therefore, we have the following:

Theorem 1. *UAP is NP-hard even if the set of users is fixed.*

To accommodate user joining and leaving, CA consists of two operations. They are AP selection at the joining phase and association re-optimization. Define vector $L = [L_1, L_2, ..., L_m]$ as the load vector. CA approximates heaviest AP load, $\max_i L_i$ by $\|L\|_p$, the ℓ_p-norm of vector L. $\|L\|_p = (\sum_i L_i^p)^{\frac{1}{p}}$. We next show that by choosing p properly, $\|L\|_p$ is at most e times of $\max_i L_i$.

Let $o = \max_i L_i$. Since o is the heaviest AP load, $\|L\|_p = (\sum_i L_i^p)^{\frac{1}{p}} \leq (mo^p)^{\frac{1}{p}} = m^{1/p}o$. That is

$$\|L\|_p \leq m^{1/p} \max_i L_i. \tag{9}$$

When $p = \ln(m)$, the factor $m^{1/p}$ achieves its minimum value e. A minimizer of $\sum_i L_i^p$ is also a minimizer of $\|L\|_p$. Therefore, we define

$$Cost = \sum_{i\in\mathbf{A}} L_i^p, \tag{10}$$

as network cost. We seek to minimize $Cost$ in our algorithm design.

When a user is joining the network, it selects the best AP to associate with according to Algorithm 1. Consider that user arrives in order. We use S_{j-1} to denote the set of previously associated users before the joining of user j. Hence, $S_j = S_{j-1} \cup j$. Denote the load of AP i after the joining of user j as $L_i(S_j)$. Define the change of the $Cost$ due to associating user j to AP n given the set of previously associated users as

$$\Delta Cost(n, j|S_{j-1}) = \sum_{i \in \mathbf{Cover}(j)} \Big((L_i(S_{j-1}) + \rho^i_{nj})^p - L^p_i(S_{j-1}) \Big).$$

Algorithm 1: Online AP selection

1 Broadcasts probe requests to discover available APs
2 Computes the marginal cost $\Delta Cost(n, j|S_{j-1})$ of associating with each AP n
3 Associates with AP n^*, such that $n^* = \arg\min_n \Delta Cost(n, j|S_{j-1})$
4 Each AP updates its current load

As shown in Line 1 of Algorithm 1, a joining user j broadcasts probe requests for AP discovery. APs in the range (i.e. APs in $\mathbf{Cover}(j)$) response with probe responses. Each AP i includes the frequency channel it uses, the number of antennas it has and its current load, $L_i(S_{j-1})$ in the response. With this information, user j computes the marginal cost and associates with the AP such that the marginal cost is minimized (Line 2 and Line 3). Since the association of a user only increases the load of APs whose interference ranges covering it, $\Delta Cost(n, j|S_{j-1})$ can be calculated distributively. The optimality of the online AP selection algorithm is stated in the following theorem.

Theorem 2. *Given that no user leaves the network, Algorithm 1 is a factor $e \log m$ online approximation algorithm. (See Appendix for proof.)*

The performance of Algorithm 1 cannot be guaranteed when the network evolves with user departure. To address it, we present association re-optimization, shown in Algorithm 2. Ideally, association re-optimization can be conducted whenever there is user departure. However, it might be costly. Hence, each user conducts association re-optimization periodically. We say an association solution is a neighbour of another solution if they are different at only one user. Given current association x, for each user j, we can construct a neighbor association by re-associating it with AP n ($x_{nj} \neq 1$). Define $\Delta Cost(n, j|x)$ as the change of $Cost$ due to re-associating user j with AP n given current association x. Re-optimization is a local search heuristic, which tries to move the current solution x to a better neighbour solution.

As shown in the algorithm, each user periodically tries to improve current association solution locally. It considers neighbour solutions formed by re-associating with a new AP and computes their costs (Line 3). Then, it re-associates to the AP such that the $Cost$ is reduced the most (Line 4). Clearly, Algorithm 2 converges. This is because whenever re-optimization is made, $Cost$ decreases and $Cost$ is lower bounded by some positive value. The performance of Algorithm 2 is stated in the following theorem.

Algorithm 2: Association re-optimization

1 **foreach** $j \in U$ **do**
2 Broadcasts probe requests to discover available APs
3 For each AP n in the range, computes $\Delta Cost(n, j|x)$
4 Associates with AP n^*, such that $n^* = \arg\min_n \Delta Cost(n, j|x)$
5 **end**

Theorem 3. *Algorithm 2 is a factor $e \log m$ approximation algorithm. (See Appendix for proof.)*

V. Distributed Joint AP Grouping and User Association

Inspired by the approximation algorithm for user association, we present in this section a distributed algorithm termed DAGA (**D**istributed Joint **A**P **G**rouping and User **A**ssociation) for joint optimization.

DAGA is a low complexity algorithm based on alternating direction optimization. It comprises user-side optimization and AP-side optimization. Based on current AP cooperation groups, user-side optimization optimizes user association according to CA (Section IV) distributively. Based on current user association, AP-side optimization periodically adjusts AP grouping distributively. For reducing control overhead and ease of implementation, User-side optimization and AP-side optimization are conducted in an asynchronized manner.

We next elaborate the AP-side optimization. Each AP group periodically negotiates with other groups and improves the common objective $Cost$ by adjusting AP grouping locally. The local adjustment can be done via the following operations: $Merge(G_i, G_j)$, $Join(i, G_j)$ and $Swap(i, j)$.

$Merge(G_i, G_j)$ operation merges APs in G_i with APs in G_j to form a new group. Let $G(i)$ be the group which AP i is currently in. In $Join(i, G_j)$ operation, AP i leaves its previous group $G(i)$ and join a new group, G_j. $Swap(i, j)$ operation swaps i and j (i.e. i leaves $G(i)$ and joins $G(j)$ at the same time j leaves $G(j)$ and joins $G(i)$). DAGA is formally presented in Algorithm 3. Since local adjustment only affects two AP groups, each operation only incurs negotiation between two groups. Therefore, control overhead for negotiation is affordable.

VI. Illustrative Simulation Results

In this section, we present our simulation results on DAGA. We discuss our simulation environment and performance metrics in Section VI-A, and illustrative simulation results in Section VI-B.

Algorithm 3: DAGA

Initially, APs are partitioned into m groups (each group contains one AP)
Each group l **repeats periodically:**
  foreach l' *in* $\mathbf{G}$, $size(G_l) + size(G_{l'}) \leq S^{max}$ **do**
    Conducts $G_l = Merge(G_l, G_{l'})$ if $Cost$ reduces
  end
  foreach i *in* G_l **do**
    foreach l' *in* $\mathbf{G}$, $size(G_{l'}) \leq S^{max} - 1$ **do**
      Conducts $Join(i, G_{l'})$ if $Cost$ reduces
    end
  end
  foreach i *in* G_l **do**
    foreach i' *not in* G_l **do**
      Conducts $Swap(i, i')$ if $Cost$ reduces
    end
  end
Each user j**:**
  Joining stage: runs Algorithm 1 to select an AP
  Repeats periodically: runs Algorithm 2 to re-optimize current association

A. Simulation Environment

We conduct simulations on a MAC-level simulator implemented using Matlab to evaluate DAGA. We simulate an enterprise WiFi network. In our simulation, APs are randomly deployed in an area (of size 200m $\times$ 200m). To create non-uniform user distribution, 70% users are uniformly distributed in a square with side 100m centered at (100m, 100m). While the other users are uniformly distributed in the whole area. Each AP operates on a channel chosen from a predefined set of orthogonal channels. Channel selection is optimized using the Tabu-search based algorithm presented in [25].

In the physical layer, we use zero-forcing beamforming for IA. Since IA requires slot synchronization between APs in the same group for channel sounding, the default DCF MAC does not naturally apply. We hence propose a MAC protocol termed Synchronized-CSMA for distributive channel access. In Synchronized-CSMA, the time slots/clocks or APs belonging to the same group are synchronized. Similar to DCF, time slots of APs in different groups are not synchronized. As we have limited the size of each group, time slots synchronization among a few number APs can be done efficiently.

Instead of using backoff timer to coordinate random channel access, Synchronized-CSMA follows the p-persistence CSMA model. After sensing an idle slot, each AP transmits with probability p, which is a system parameter. To allow APs within the same group to make use of IA, AP in the same group sensing an idle slot decide to transmit or not transmit synchronously according to persistence p. For example, AP 1 and AP 2 in group A sense idle channel. They draw a random number r from $[0, 1]$. Both of them transmit if $r \leq p$; give up the transmission opportunity otherwise.

In such way, AP in the same group access channel synchronously and hence can perform channel sounding and user selection jointly. APs winning contention select users for downlink transmission according to the longest-queue-first user selection algorithm. To reduce MAC overhead, APs transmit for TO number or time slots, which is referred to as transmission opportunity.

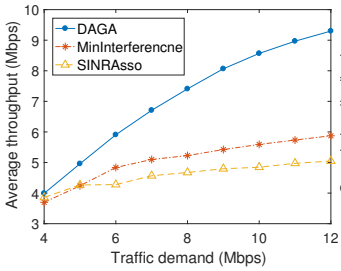

Fig. 3. Throughput vs. offered load.

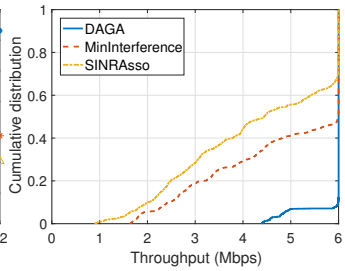

Fig. 4. Throughput distribution.

Unless otherwise stated, we use the following as our baseline parameters. We use log-distance path loss model with reference distance 1 meter, reference loss 46.678 dBm and loss exponent 3. The noise power of the environment is -101 dBm. There are 3 orthogonal spectrum channels in the network, each is with bandwidth of 20 MHz. The number of antennas at an AP is drawn from a normal distribution with mean of 4 and variance of 1. The number of APs and Users are 30 and 200 respectively. Each user receives downlink traffic from the associated AP. Default traffic demand is 6Mbps. Maximum group size is 4. Slot time is 10μs. A transmission opportunity contains 10 slots, i.e., $TO = 10$.

We are interested in the following performance metrics: 1) *Throughput:* It is the MAC-layer throughput, calculated as the number of bits received divided by transmission duration. 2) *Jain's fairness index:* It measures the fairness of throughput given by a set of n users. It is calculated as $(\sum_{i=1}^{n} T_i)^2/(n\sum_{i=1}^{n} T_i^2)$, where T_i is the throughput of the ith user.

We compare DAGA with the following approaches:

1) *Proportional fairness association based on SINR (***SINRAsso***) [5]:* It represents a body of work in the literature. It does not consider spatial domain IA or time domain interference avoidance. In modeling user received SINR, APs are assumed to transmit simultaneously. Hence, the signal power from all APs except the associated one is considered as interference. Each user selects the best AP in an online manner according to SINR and the load sharing of APs.

2) *Minimum interference AP grouping (***MinInterference***):* In the scheme, users are first associated with the AP from which it receives the strongest signal. Based on the links formed by the association solution, Algorithm 1 presented in [22] then clusters APs into cooperation groups to minimize inter-group interference.

B. Illustrative Simulation Results

We plot the average user throughput versus different per-user traffic demand in Figure 3. The throughput of all schemes

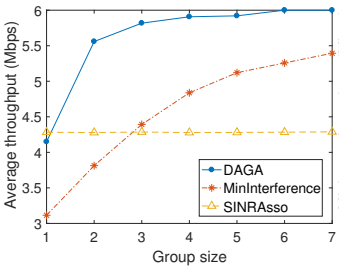

Fig. 5. Throughput vs. group size.

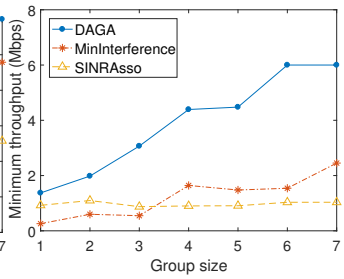

Fig. 6. Minimum throughput vs. group size.

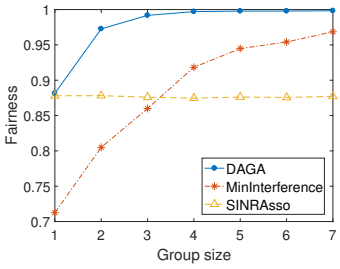

Fig. 7. Fairness vs. group size.

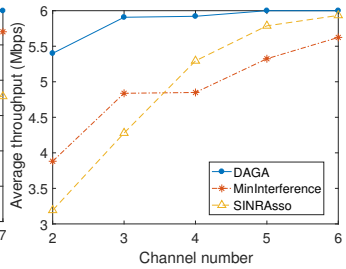

Fig. 8. Throughput vs. channel number.

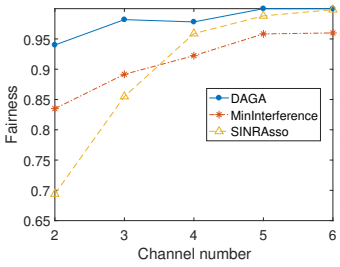

Fig. 9. Fairness vs. channel number.

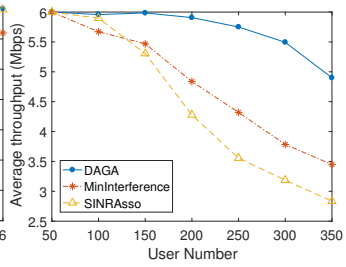

Fig. 10. Throughput vs. user number.

first increases with traffic demand, then flats off as the traffic saturates the network. SINRAsso performs the worst. DAGA outperforms SINRAsso mainly due to the following reasons: 1) SINRAsso does not take interference coordination schemes into consideration. It assumes all APs transmit simultaneously and signal power from neighbour APs is modeled as interference. When the network is dense, SINRAsso tends to associate users with the closest AP to improve received SINR, and hence can not achieve load balancing. However, in the system under study, interference is either coordinated with IA or overcome by time domain avoidance. This justifies that user association needs to be aware of interference coordinating schemes applied. 2) It does not leverage IA to allow interfering links to be scheduled simultaneously. DAGA outperforms MinInterference because MinInterference aims to minimize overall interference and hence maximizing sum link rate. It does not consider load balance among different groups. As a result, a lot of users may be grouped into one group lead to heavy traffic load at APs in the group. Furthermore, MinInterference does not leverage user association to balance AP load or optimize network topology.

The throughput CDF of different schemes is plotted in Figure 4. Most users in DAGA enjoy high throughput, which is substantially better than MinInterference and SINRAsso. A large portion of users in SINRAsso suffer from low throughput allocation (the throughput of 50% users is lower than 5 Mbps). A significant number of users in MinInterference receive low throughput allocation because the grouping algorithm aims to minimize overall interference. It does not consider load balance among different groups.

Figure 5 plots the average user throughput versus maximum grouping size. We get higher throughput as we allow larger group size. This is because more APs are able to cooperate to reduce interference such that more users can be scheduled in a time slot. DAGA outperforms the comparison schemes. The throughput increment slows down as maximum group size increases to a certain point. This is because we consider the case of finite traffic demand, i.e., the traffic injected by each user to the network is 6Mbps in our study. The performance of SINRAsso remains the same under different group size because it does not leverage AP cooperation.

In Figure 6, we compare worst-case user throughput of DAGA versus different group size with other schemes. As expected, the worst-case user throughput in all schemes increases with maximum group size as more cooperation help to reduce inter-AP interference. DAGA outperforms the comparison schemes due to the gain of joint optimization.

The Jains fairness versus different group size of different schemes is shown in Figure 7. User throughput fairness improves with maximum group size. The fairness of MinInterference increases drastically with group size, which is substantially better than that of SINRAsso when the group size is beyond a certain value. It is because SINRAsso does not perform interference alignment for users covered by multiple APs. Without IA, users in the overlapping coverage suffer from heavy channel contention. Therefore, they are scheduled with lower probability.

We plot the average throughput of different schemes versus the number of available channels in Figure 8. The throughput of all schemes increases with the number of channels because channel contention/interference reduces greatly as frequency resource increase.

In Figure 9, we compare the Jain's fairness of DAGA under different channel resource with other comparison schemes. Fairness increases with channel number. DAGA outperforms SINRAsso when there is only a few number of channels. This shows that without the awareness of IA, SINRAsso associates users with the closest AP to achieve higher SINR when there are a lot of APs sharing the same channel. Therefore, it cannot achieve load balancing. When the number of channels is beyond a certain point (4 in our study), SINRAsso achieves similar fairness as DAGA.

We show average user throughput versus different number of users in Figure 10. Average user throughput decreases with the number of users. As the frequency resource of the network

is limited, each user shares less resource as user number getting larger.

VII. Conclusion

We consider general MIMO networks, such as WiFi networks or small cell networks, where APs are equipped with heterogeneous number of antennas and users are non-uniformly distributed. A hybrid interference coordination approach is used to mitigate interference, which partitions APs into multiple cooperation groups no larger than a certain size. Intra-group interference is nullified with IA, while inter-group interference is overcome using traditional orthogonal access based approaches such as time or spectral division. Unbalanced AP load due to non-uniform user traffic is addressed using association control. We study the joint optimization of AP grouping and user association to minimize heaviest AP Load. To our best knowledge, this is the first piece of work addressing this problem.

We formulate the optimization problem and show that it is NP-hard. We propose DAGA, a fully distributed algorithm based on alternating direction optimization. It alternates between association optimization and AP grouping optimization. Given AP grouping, DAGA is a factor $e \log m$ approximation algorithm for user association. Using extensive simulation, we show that DAGA outperforms other state-of-the-art schemes.

References

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

[2] H. Yu, O. Bejarano, and L. Zhong, "Combating inter-cell interference in 802.11 ac-based multi-user mimo networks," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 141–152.

[3] G. Athanasiou, P. C. Weeraddana, C. Fischione, and L. Tassiulas, "Optimizing client association for load balancing and fairness in millimeter-wave wireless networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 836–850, 2015.

[4] F. Xu, C. C. Tan, Q. Li, G. Yan, and J. Wu, "Designing a practical access point association protocol," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[5] W. C. Ao and K. Psounis, "Approximation algorithms for online user association in multi-tier multi-cell mobile networks," *IEEE/ACM Transactions on Networking*, 2017.

[6] Z. Chen, Q. Xiong, Y. Liu, and C. Huang, "A strategy for differentiated access service selection based on application in WLANs," in *IEEE INFOCOM WKSHPS*, 2014, pp. 317–322.

[7] H. Kim, W. Lee, M. Bae, and H. Kim, "Wi-fi seeker: a link and load aware ap selection algorithm," *IEEE Transactions on Mobile Computing*, 2016.

[8] T. Sun, Y. Zhang, and W. Trappe, "Improving access point association protocols through channel utilization and adaptive probing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1157–1167, 2016.

[9] Y. Zhang, D. Bethanabhotla, T. Hao, and K. Psounis, "Near-optimal user-cell association schemes for real-world networks," in *Information Theory and Applications Workshop (ITA), 2015*. IEEE, 2015, pp. 204–213.

[10] W. Li, S. Wang, Y. Cui, X. Cheng, R. Xin, M. Al-Rodhaan, and A. Al-Dhelaan, "Ap association for proportional fairness in multirate WLANs," *IEEE/ACM Trans. Netw*, vol. 22, no. 1, pp. 191–202, 2014.

[11] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless lans using association control," in *Proc. ACM MobiCom*, 2004, pp. 315–329.

[12] Y.-C. Hsu, K. C.-J. Lin, and W.-T. Chen, "Client-AP association for multiuser mimo networks," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 2154–2159.

[13] T. Van Chien, E. Björnson, and E. G. Larsson, "Joint power allocation and user association optimization for massive mimo systems," *arXiv preprint arXiv:1601.02436*, 2016.

[14] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, "Cell association and interference coordination in heterogeneous lte-a cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1479–1489, 2010.

[15] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "Optimal user-cell association for massive mimo wireless networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1835–1850, 2016.

[16] N. Wang, E. Hossain, and V. K. Bhargava, "Joint downlink cell association and bandwidth allocation for wireless backhauling in two-tier hetnets with large-scale antenna arrays," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3251–3268, 2016.

[17] R. Tresch and M. Guillaud, "Clustered interference alignment in large cellular networks," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*. IEEE, 2009, pp. 1024–1028.

[18] J. Park, N. Lee, and R. W. Heath, "Cooperative base station coloring for pair-wise multi-cell coordination," *IEEE Transactions on Communications*, vol. 64, no. 1, pp. 402–415, 2016.

[19] S. W. Peters and R. W. Heath, "User partitioning for less overhead in mimo interference channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, pp. 592–603, 2012.

[20] K. Hosseini, W. Yu, and R. S. Adve, "Cluster based coordinated beamforming and power allocation for mimo heterogeneous networks," in *Information Theory (CWIT), 2013 13th Canadian Workshop on*. IEEE, 2013, pp. 96–101.

[21] R. Brandt, R. Mochaourab, and M. Bengtsson, "Distributed long-term base station clustering in cellular networks using coalition formation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 3, pp. 362–375, 2016.

[22] S. Chen and R. S. Cheng, "Clustering for interference alignment in multiuser interference network," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2613–2624, 2014.

[23] H. Zeng, Y. Shi, Y. T. Hou, W. Lou, S. Kompella, and S. F. Midkiff, "An analytical model for interference alignment in multi-hop mimo networks," *IEEE transactions on mobile computing*, vol. 15, no. 1, pp. 17–31, 2016.

[24] B.-X. Wu, K. C.-J. Lin, K.-C. Hsu, and H.-Y. Wei, "Hybridcast: Joint multicast-unicast design for multiuser mimo networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 1724–1732.

[25] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1459–1473, 2008.

[26] I. Caragiannis, "Better bounds for online load balancing on unrelated machines," in *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2008, pp. 972–981.

Appendix

Proof of Theorem 2 (See Page 5.)

Proof. This proof is built on Theorem 3.1 of [26], which studies assigning jobs to machines. The major difference is that assigning a job to a machine only increases the load on the machine. However, associating a user to an AP incurs load on multiple APs. Consider users arrive to the network in the order $1, 2, ..., n$. Users are labeled by the order they arrive (i.e. the jth user is labeled as user j). Let x be the association solution given by Algorithm 1 and x^* be the optimal association solution. $x^*_{ij} = 1$ if optimal algorithm associates j with i; Otherwise, $x^*_{ij} = 0$. After the joining of user j, Algorithm 1 produces $L_i(S_j)$ amount of load on AP i.

At the step of associating user j to the network, we have

$$\sum_i L_i^p(S_j) - L_i^p(S_{j-1})$$

$$\leq \sum_i \left(L_i(S_{j-1}) + \sum_n x^*_{nj} \rho^i_{nj} \right)^p - L_i^p(S_{j-1})$$

$$\leq \sum_i \left(L_i(S_n) + \sum_n x^*_{nj} \rho^i_{nj} \right)^p - L^p_i(S_n) \quad (11)$$

The first inequality is due to the fact that we assign j to the AP such that the increase of $Cost$ is minimized. Function $f(x) = (x+a)^p - x^p$ is non-decreasing in $[0, \infty)$ for $a \geq 0$ and $p \geq 1$. Since $L_i(S_n) \geq L_i(S_{j-1})$, we have the second inequality. The network cost of Algorithm 1 is calculated as

$$\sum_i L^p_i(S_n) \quad (12)$$

$$= \sum_i \sum_j \left(L^p_i(S_j) - L^p_i(S_{j-1}) \right) \quad (13)$$

$$= \sum_j \sum_i \left(L^p_i(S_j) - L^p_i(S_{j-1}) \right) \quad (14)$$

$$\leq \sum_j \sum_i \left((L_i(S_n) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(S_n) \right) \quad (15)$$

$$= \sum_i \sum_j \left((L_i(S_n) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(S_n) \right) \quad (16)$$

We associate users one by one in n steps. $L^p_i(S_j) - L^p_i(S_{j-1})$ is the load increment on AP i in the jth step. We have (13) from (12) because the final load of an AP is just the summation of load increment over all steps. Using (11), we get (15).

Let $p \geq 1, y \geq 0$, and $a_j \geq 0, \forall j$. We have $\sum_j ((y+a_j)^p - y^p) \leq (y + \sum_j a_j)^p - y^p$. Therefore, we get the following from (16)

$$\sum_i L^p_i(S_n)$$

$$\leq \sum_i \left((L_i(S_n) + \sum_j \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(S_n) \right) \quad (17)$$

$$= \sum_i (L_i(S_n) + L^*_i)^p - \sum_i L^p_i(S_n) \quad (18)$$

$$\leq \left((\sum_i L^p_i(S_n))^{1/p} + (\sum_i L^{*p}_i)^{1/p} \right)^p - \sum_i L^p_i(S_n) \quad (19)$$

We get (19) from (18) by applying Minkowski Inequality to $\sum_i (L_i(S_n) + L^*_i)^p$. $(\sum_i L^p_i(S_n))^{1/p}$ is the ℓ_p-norm of AP load produced by Algorithm 1, while $(\sum_i L^{*p}_i)^{1/p}$ is the ℓ_p-norm of AP load given by the optimal algorithm. The approximation ratio of minimizing $\|L\|_p$ is denoted as c, $c = \frac{(\sum_i L^p_i(S_n))^{1/p}}{(\sum_i L^{*p}_i)^{1/p}}$.

Dividing both sides of (19) by $(\sum_i L^{*p}_i)^{1/p}$, we get $2c^p \leq (c+1)^p$. Therefore, $c \leq \frac{1}{2^{1/p}-1} \leq \frac{p}{\ln 2}$. Algorithm 1 produces a factor $\frac{p}{\ln 2}$ approximation to the ℓ_p-norm minimization. Moreover, ℓ_p-norm minimization is a factor $m^{1/p}$ approximation to minimize the heaviest AP load (Inequality (9)). Hence, Algorithm 1 is a factor $m^{1/p} \frac{p}{\ln 2}$ online approximation algorithm for minimizing the heaviest AP load, $max_i L_i$. By choosing $p = \ln m$, $m^{1/p} \frac{p}{\ln 2} = e \log m$. □

Proof of Theorem 3 (See Page 5.)

Proof. We label the set of users U in the network as $1, 2, ..., n$ in arbitrary order. Let U_j be the set including users $1, 2, ..., j$. Users in U_j are associated to the network according to x. The load they create on AP i is $L_i(U_j)$.

Set $U - j$ includes all users but user j. The load on AP i caused by users in $U - j$ is $L_i(U - j)$. When Algorithm 2 converges, we get a locally optimal solution x. That is for each user j, we cannot reduce $Cost$ by re-associating it to a new AP. Denote the optimal association as x^*. We have,

$$\sum_i \left(L^p_i(U) - L^p_i(U-j) \right)$$
$$\leq \sum_i \left((L_i(U-j) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(U-j) \right), \forall j \in U. \quad (20)$$

Let a^*_j be the AP associated by user j in the optimal solution. The above inequality must be true. Otherwise, the algorithm will re-associate user j to a^*_j as this reduces network cost.

$U - j = \{1, 2, ..., j-1, j+1, ..., n\}$ and $U_{j-1} = \{1, 2, ..., j-1\}$. Therefore, $L^p_i(U-j) \geq L^p_i(U_{j-1}) \geq 0$. Since function $f(x) = (x+a)^p - x^p$ is non-decreasing in $[0, \infty)$ for $a \geq 0$ and $p \geq 1$,

$$\sum_i \left(L^p_i(U) - L^p_i(U-j) \right)$$
$$= \sum_i \left((L_i(U-j) + \sum_n x_{nj} \rho^i_{nj})^p - L^p_i(U-j) \right)$$
$$\geq \sum_i \left((L_i(U_{j-1}) + \sum_n x_{nj} \rho^i_{nj})^p - L^p_i(U_{j-1}) \right)$$
$$= \sum_i \left(L^p_i(U_j) - L^p_i(U_{j-1}) \right) \quad (21)$$

$$\sum_i \left((L_i(U-j) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(U-j) \right)$$
$$\leq \sum_i \left((L_i(U) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(U) \right) \quad (22)$$

Replace the LHS (left hand side) of (20) by (21) and the RHS by (22). We have

$$\sum_i \left(L^p_i(U_j) - L^p_i(U_{j-1}) \right) \quad (23)$$
$$\leq \sum_i \left((L_i(U) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(U) \right), \forall j$$

Summing (23) overall all j, we have

$$\sum_i L^p_i(U) = \sum_i \sum_j \left(L^p_i(U_j) - L^p_i(U_{j-1}) \right)$$
$$\leq \sum_i \sum_j \left((L_i(U) + \sum_n x^*_{nj} \rho^i_{nj})^p - L^p_i(U) \right) \quad (24)$$

Comparing (24) with (16), they are the same if we replace S_n by U. Applying the techniques used for proving Theorem 1, we get the desired result.

□