



Research Repository UCD

Title	Noncoherent Joint Transmission Beamforming for Dense Small Cell Networks: Global Optimality, Efficient Solution and Distributed Implementation
Authors(s)	Vu, Quang-Doanh, Tran, Le-Nam, Juntti, Markku
Publication date	2020-09
Publication information	Vu, Quang-Doanh, Le-Nam Tran, and Markku Juntti. "Noncoherent Joint Transmission Beamforming for Dense Small Cell Networks: Global Optimality, Efficient Solution and Distributed Implementation." IEEE, September 2020. https://doi.org/10.1109/TWC.2020.2998067 .
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/11904
Publisher's statement	© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/TWC.2020.2998067

Downloaded 2025-12-04 22:41:59

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

Noncoherent Joint Transmission Beamforming for Dense Small Cell Networks: Global Optimality, Efficient Solution and Distributed Implementation

Quang-Doanh Vu, *Member, IEEE*, Le-Nam Tran, *Senior Member, IEEE*, and
Markku Juntti, *Fellow, IEEE*

Abstract

We investigate the coordinated multi-point noncoherent joint transmission (JT) in dense small cell networks. The goal is to design beamforming vectors for macro cell and small cell base stations (BSs) such that the weighted sum rate of the system is maximized, subject to a total transmit power at individual BSs. The optimization problem is inherently nonconvex and intractable, making it difficult to explore the full potential performance of the scheme. To this end, we first propose an algorithm to find a globally optimal solution based on the generic monotonic branch reduce and bound optimization framework. Then, for a more computationally efficient method, we adopt the inner approximation (InAp) technique to efficiently derive a locally optimal solution, which is numerically shown to achieve near-optimal performance. In addition, for decentralized networks such as those comprising of multi-access edge computing servers, we develop an algorithm based on the alternating direction method of multipliers,

This work was supported in part by the Academy of Finland under the projects “Flexible Uplink-Downlink Resource Management for Energy and Spectral Efficiency Enhancing in Future Wireless Networks (FURMESFuN)” under Grant 310898, and “6Genesis Flagship” under Grant 318927. This publication has emanated from research supported in part by a Grant from Science Foundation Ireland under Grant number 17/CDA/4786.

Quang-Doanh Vu, and Markku Juntti are with Centre for Wireless Communications, University of Oulu, FI-90014, Finland. Email: {doanh.vu, markku.juntti}@oulu.fi.

L.-N. Tran is with School of Electrical and Electronic Engineering, University College Dublin, Ireland. Email: nam.tran@ucd.ie).

Part of this work was presented at IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), Brighton, United Kingdom, May 12-17, 2019 [1].

which distributively implements the InAp-based solution. Our main conclusion is that the noncoherent JT is a promising transmission scheme for dense small cell networks, since it can exploit the densification gain, outperforms the coordinated beamforming, and is amenable to distributed implementation.

Index Terms

Dense small cell networks, noncoherent joint transmission, weighted sum rate, multi-access edge computing, distributed implementation, branch reduce and bound, inner approximation, alternating direction method of multipliers.

I. INTRODUCTION

The rapid growth in the number and the diverse requirements of wireless communications applications has presented the paramount challenge of wirelessly transmitting huge volumes of data for the upcoming mobile networks. It is predicted that the total mobile traffic will be five times higher by 2023 compared to 2018 [2]. In addition, the next generation of mobile networks is going to introduce various service categories to support diverse communication requirements, e.g., enhanced mobile broadband, massive machine type communications (mMTC), and ultra-reliable low-latency communications (URLLC) [3]. Dense small cell deployment is one promising technology to enable the new services [4]. With densification of low-cost base stations (BSs), the existing spectrum is exploited efficiently by the spatial reuse, and the energy efficiency is enhanced due to the short-range wireless transmission [5]. Furthermore, the proximity of the cells to the users can support low latency services as well as guarantee quality of experience (QoE) [6].

Designing radio access networks (RANs) has been progressing significantly during the recent few years. The centralized RAN (CRAN) architecture moves the baseband processing functionalities of the conventional base station (BS) to a central location called baseband unit (BBU) pool [7], [8]. This fully centralized architecture exploits powerful cloud computing capabilities for resource management. However, it requires local information, e.g., channel state information (CSI), to be gathered at the centralized BBU pool [9], which needs a significant cost invested in the fronthaul network and might result in high latency [10]. In order to overcome these shortcomings, the concept of multi-access edge computing (MEC), named by the European Telecommunications Standards Institute (ETSI), has recently been introduced [10]–[15]. The technology deploying the computing, storage and networking resources (called MEC servers)

across networks allows data to be stored and processed locally [10], [14], [15]. As such, the networks with MEC technique are sort of decentralized architecture [12], [13]. In dense small cell networks, MEC servers could be co-located with selected small cell BSs [10].

In dense small cell networks, the BSs are close to each other. Thus, efficiently managing inter-cell interference is apparently one of the keys for a successful implementation [16]. A common approach is to use coordinated multi-point (CoMP) strategies. The simplest form of CoMP is coordinated beamforming (CB) where a specific user receives data from only one BS, while the interference caused by other BSs is reduced by the cooperation of the involved BS [17]. The most advanced CoMP strategy is *coherent* joint transmission (JT) where data for a user is available at multiple BSs, and the BSs collaborate to create a large virtual multiple-input multiple-output (MIMO) system in order to maximally exploit array gain [18]. However, coherent JT requires strict synchronization among BSs (0.5 microsecond accuracy [19]). This requirement remains a main challenge for a practical implementation of coherent JT [20], even in centralized architecture systems (i.e., CRAN) where the synchronization is improved (compared to the conventional RAN) [19]. Recently, *noncoherent joint transmission* has received growing attention [21]–[24], since it requires not as strict synchronization accuracy as compared to coherent JT [21]–[24]. We note that the term ‘noncoherent’ is used herein in the context of joint transmission referring to the signal processing coordination among BSs. It does not refer to the classical notions of noncoherent communications or data detection, in which neither the carrier phase is available to the receiver [25], nor the instantaneous channel is known at the receiver [26], [27]. In our paper, noncoherent JT refers to the scenario where users still receive data from multiple BSs, but the data is encoded independently at individual BSs [21], and users apply successive interference cancellation to decode its information where the information from a BS is decoded with the signal from the remaining BSs is treated as noise [23]. As such, noncoherent JT is expected to require BS synchronization at the same level as CB (3 microsecond accuracy [19]).

A. Contributions

The above discussion motivates us to investigate the achievable performance of the noncoherent JT technique in the context of dense small cell networks where the BSs collaborate to serve a set of users. The target is to design beamforming vectors at BSs so that the weighted sum rate (WSR) is maximized under the constraints on maximum transmit power at each of the BSs. We consider the WSR as the objective to be maximized, because it is general enough to encompass

other performance measures such as spectral efficiency and the guaranteed quality of services for the users (via appropriate weights) as special cases [28]. The contributions of this paper are as follows:

- *Globally optimal solution:* We first find the optimal beamforming vectors to fully understand the potential performance of the noncoherent JT. The nonconvexity and intractability of many design problems related to the WSR maximization have been widely known in the literature [29]. To this end, we develop an algorithm based on the branch reduce and bound (BRnB) monotonic optimization framework which globally solves the considered problem [30].
- *Computationally efficient solution:* We develop a low-complexity suboptimal iterative method based on the well-known inner approximation (InAp) framework [31], [32], which is provably convergent and efficiently solves the WSR maximization problem. In each iteration, only a conic quadratic program (CQP) needs to be solved. Also, we numerically demonstrate the fast convergence and the near-optimal performance of the suboptimal solution.
- *Distributed implementation:* We subsequently develop a distributed implementation of the efficient solution for decentralized architecture in dense small cell networks deploying the MEC servers. Particularly, motivated by the appreciated success of the alternating direction method of multipliers (ADMM) in designing distributed algorithms reported in recent publications [33]–[36], we rely on this mathematical tool to decompose the convex approximated problems (i.e., the CQP) obtained by the InAp-based method into subproblems, which can be solved locally at the MEC servers. As such, the beamforming vectors can be computed at the MEC servers using local information.

We provide extensive numerical results to evaluate the efficiency of the proposed methods. In particular, we conclude that the noncoherent JT is suitable for the decentralized architecture dense small cell networks due to the fact that it has the ability of exploiting densification gain, and is convenient for being implemented distributively.

B. Related Works

There is a large portion of related works in the subject of small cell networks but mostly focusing on coherent JT. For example, the authors in [37] and [38] designed precoding for minimizing power and maximizing energy efficiency, respectively. In CRAN based networks, coherent JT was considered in [39], [40], [41], and [42] for WSR maximization, energy efficiency

maximization, power minimization, and multi-objective of spectral and power efficiency maximization, respectively. These previous works implicitly assume a strict requirement on network synchronization.

The CB has been investigated extensively. The common approaches for the WSR problem in CB systems include weighted minimum mean square error (WMMSE) [43] and InAp (or successive convex approximation) [44], which were numerically shown to achieve near-optimal performance [44]. We will see in the next section that the numerator of the signal-to-interference-plus-noise ratio (SINR) expression of the noncoherent JT is a sum of multiple quadratic functions, which is different from the SINR expression of the CB. Consequently, the solutions developed for CB systems are not readily suitable for noncoherent JT. Particularly, it remains to be seen how the WMMSE can be applied to the noncoherent JT, because the approach of introducing the auxiliary variables in the CB is no longer useful [23]. Also, the InAp-based solution developed in [44] cannot be directly applied to the noncoherent JT, since it was derived based on the phase rotation technique, which does not lead to a tractable formulation in the noncoherent JT context. The zero forcing (ZF) technique can be applied to the noncoherent JT [24], but ZF may be infeasible for dense small cell networks, because small cell BSs are usually equipped with a few antennas. Furthermore, a user in a CB system receives desired signals from only one BS while in the noncoherent JT, it receives desired signals from multiple BSs. Thus, the existing distributed algorithms for the CB cannot be straightforwardly applied to the noncoherent JT.

Noncoherent JT has received growing attention, since it requires less strict network synchronization accuracy compared to the coherent counterpart [21]–[24]. In [21], beamforming vectors at the BSs were designed for minimizing the power consumption subject to users' minimum data rate. In [24], noncoherent JT was studied for the two problems: (i) power minimization subject to users' minimum data rate, and (ii) weighted max-min fairness. Therein, each BS is equipped with a massive number of antennas and simple beamforming schemes, i.e., ZF and maximum ratio transmission (MRT), are used. As discussed above, ZF is not a feasible approach for small cell systems. It is noted that, different from the power minimization problem, applying MRT for a WSR maximizing scheme does not lead to a tractable problem. Noncoherent JT design for minimizing weighted power consumption with imperfect channel state information was considered in [22]. A heuristic beamforming design for maximizing the WSR under the limited fronthaul capacity for CRAN networks was proposed in [23], which showed that noncoherent JT might outperform coherent JT in the regime of low fronthaul capacity. Generally, for power

minimization problems, the optimal beamforming vectors for noncoherent JT can be exactly found since their semidefinite relaxation (SDR) versions are tight and convex [21]. However, for the WSR maximization, its SDR is still intractable.

In summary, the full potential performance and useful insights into the design of noncoherent JT in dense small cell networks in terms of WSR maximization has not been previously well studied. Moreover, there is a lack of an efficient distributed algorithm implementing noncoherent JT in decentralized architecture networks. These objectives are the main focus in this paper.

C. Organization and Notations

The rest of the paper is organized as follows. Section II describes the system model and the problem formulation of designing noncoherent JT beamforming for maximizing WSR. Section III presents a globally optimal solution of the problem. An efficient solution is provided in Section IV followed by its distributed implementation presented in Section V. Numerical results and discussions are provided in Section VI. Finally, Section VII concludes the work.

Notations: Bold lower and upper case letters represent vectors and matrices, respectively; $\|\cdot\|_p$ represents the ℓ_p norm; $|\cdot|$ is the absolute value of the argument; $\mathbb{C}^{x \times y}$ represents the space of complex matrices of dimensions given in the superscript; \mathbb{S}_+^x denotes the space of symmetric positive semidefinite matrices; $\mathcal{CN}(0, a)$ denotes a complex Gaussian random variable with zero mean and variance a ; $\text{Re}\{\cdot\}$ represents real part of the argument. Notation \mathbf{e}_i denotes the i th conventional basis vector, i.e., the vector such that $e_i = 1$ and $e_j = 0, \forall j \neq i$. \mathbf{X}^T and \mathbf{X}^H stand for the transpose and the Hermitian transpose of \mathbf{X} , respectively. We use “MATLAB notation” $\text{blkdiag}\{\cdot\}$ which represents block diagonal matrix.

II. SYSTEM MODEL

A. Signal Transmission

We consider a region covered by a macro cell BS and a set of K small cell BSs shown in Fig. 1. Let us denote by $\mathcal{K} = \{1, 2, \dots, K+1\}$ the set of all BSs where $\{1\}$ refers to the macro BS and the rest the small cell BSs. BS k is equipped with M_k antennas. The BSs simultaneously serve a set of N single-antenna users, denoted by $\mathcal{N} = \{1, 2, \dots, N\}$ under the same frequency band. Let B_k and U_i denote BS k and user i , respectively. Herein, we assume that the BSs collaborate using noncoherent JT, i.e., the information for a specific user is encoded independently at individual BSs [21]. Particularly, let s_{ik} and $\mathbf{v}_{ik} \in \mathbb{C}^{M_k \times 1}$ be the normalized symbol and the beamforming

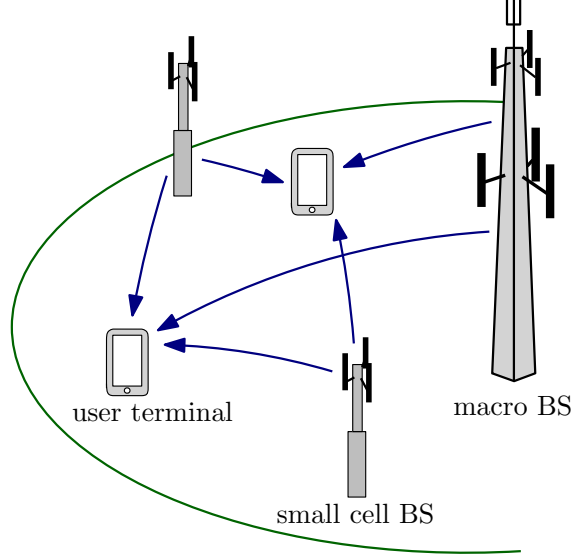


Figure 1. Small cell deployment system model.

vector at B_k for U_i , respectively. Let $\mathbf{h}_{ik} \in \mathbb{C}^{1 \times M_k}$ (row vector) be the channel between B_k and U_i , which is assumed to be perfectly known. The signal received at U_i under the assumption of flat channels is given by

$$r_i = \sum_{k \in \mathcal{K}} \mathbf{h}_{ik} \mathbf{v}_{ik} s_{ik} + \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} \mathbf{h}_{ik} \mathbf{v}_{jk} s_{jk} + z_i \quad (1)$$

where $z_i \sim \mathcal{CN}(0, \sigma_i^2)$ is the additive white Gaussian noise. The first and second sum in the right side of (1) are the desired signal and the interference, respectively. The users are assumed to use successive interference cancellation technique to detect its own signal and treat signal of other users as noise. Thus the *effective* (or aggregated) SINR at U_i can be written as [21]¹

$$\gamma_i(\{\mathbf{v}_{ik}\}) = \frac{\sum_{k \in \mathcal{K}} |\mathbf{h}_{ik} \mathbf{v}_{ik}|^2}{\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2}. \quad (2)$$

We note that $\{\gamma_i(\{\mathbf{v}_{ik}\})\}_{i \in \mathcal{N}}$ are achieved without phase synchronization between BSs. We also remark that $\{\gamma_i\}_{i \in \mathcal{N}}$ are the aggregated instantaneous SINR, i.e., the total information received at U_i is $\log(1 + \gamma_i)$ [21]. The reader is referred to [23] for the derivation of $\gamma_i(\{\mathbf{v}_{ik}\})$.

¹We note that the decoding order has no impact on γ_i [23].

B. Problem Formulation

We aim at designing beamforming vectors $\{\mathbf{v}_{ik}\}_{i,k}$ to maximize the WSR under the constraints of the transmit power budget at the BSs. Mathematically, the problem reads

$$\underset{\{\mathbf{v}_{ik}\}}{\text{maximize}} \sum_{i \in \mathcal{N}} w_i \log(1 + \gamma_i(\{\mathbf{v}_{ik}\})) \quad (3a)$$

$$\text{subject to } \sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K} \quad (3b)$$

where $w_i > 0$ is the priority of U_i , and P_k is the maximum transmit power available to B_k . Note that the SINR in (2) is nonconvex with $\{\mathbf{v}_{ik}\}_{i,k}$, which makes problem (3) intractable [21].

C. Centralized and Distributed Architectures of Small Cell Networks

The noncoherent JT technique can be deployed on the centralized or distributed platforms as illustrated in Fig. 2. The centralized one mainly refers to the CRANs where the baseband processing functionalities of the BSs are centralized at the BBU pool [7], [8]. It requires channel vector \mathbf{h}_{ik} to be gathered at the BBU pool for all $i \in \mathcal{N}, k \in \mathcal{K}$, where all beamforming vectors $\{\mathbf{v}_{ik}\}_{i \in \mathcal{N}, k \in \mathcal{K}}$ are calculated. After baseband processing, the signal is sent to the BSs (remote radio head to be precise) via fronthaul links before being transmitted to users via the wireless interface. We herein focus on the performance of noncoherent JT over wireless channels. Thus, we suppose that the capacity of the fronthaul links are sufficiently large and not forming the bottleneck.

Distributed platform refers to the MEC architecture [10]–[15], which is introduced to support low-latency services. Here, multiple MEC servers can be deployed across the networks to store and process data locally. In dense small cell networks, baseband processing function of one or several BSs located close to each other can be gathered at a MEC server to exploit the server's computing capacity. A MEC server can be physically co-located with a selected BS [10]. To efficiently manage interference of dense small cells, the MEC servers should cooperate with each other in baseband processing.

We herein consider both centralized and distributed network architectures. For the latter, we suppose that there is a set of D MEC servers denoted by $\mathcal{D} = \{1, \dots, D\}$. MEC server d handles the baseband processing functionality of a set of BSs denoted by $\mathcal{K}_d, \mathcal{K}_d \subset \mathcal{K}$. We also suppose that each of the MEC servers can serve multiple BSs, and each of the BSs is served by only one MEC server, i.e., $|\mathcal{K}_d| \geq 1, \mathcal{K}_d \cap \mathcal{K}_{d'} = \emptyset, \forall d \neq d'$, and $\bigcup_{d \in \mathcal{D}} \mathcal{K}_d = \mathcal{K}$. Thus, beamforming vectors

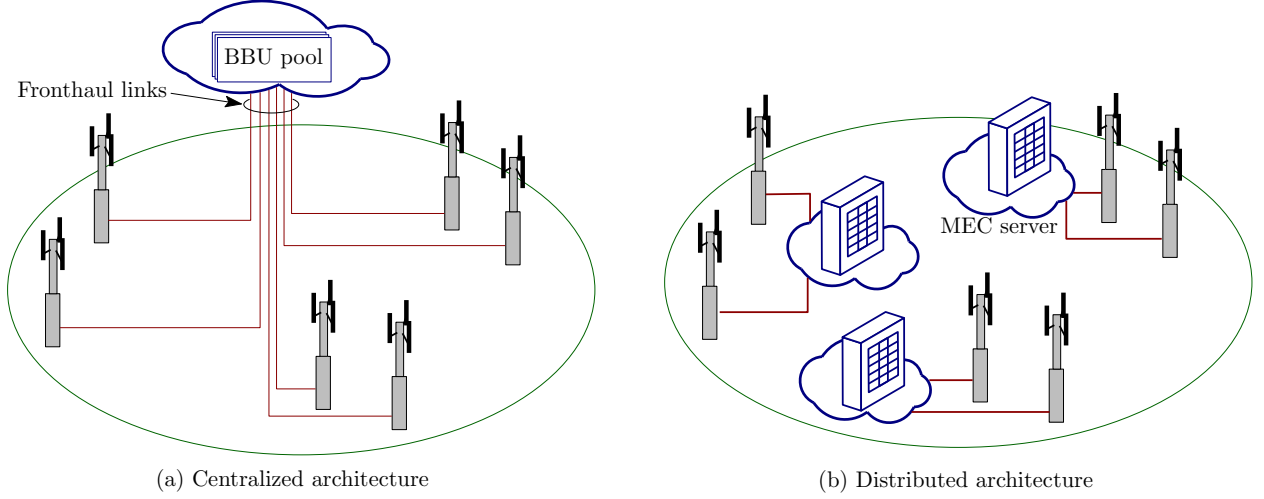


Figure 2. Centralized and distributed network architectures.

$\{\mathbf{v}_{ik}\}_{i \in \mathcal{N}, k \in \mathcal{K}_d}$ can be locally computed at MEC server d if the local CSI, i.e., $\{\mathbf{h}_{ik}\}_{i \in \mathcal{N}, k \in \mathcal{K}_d}$, is available.

III. GLOBALLY OPTIMAL SOLUTION TO (3): A BRnB ALGORITHM

In this section, we develop an algorithm which globally solves (3) based on the BRnB monotonic optimization framework [30]. The purpose of finding a global solution to (3) is twofold: (i) exploring the best achievable performance of the noncoherent JT technique in small cell networks, and (ii) benchmarking against the efficient suboptimal solution to be presented. We start by rewriting (3) as

$$\underset{\{\mathbf{v}_{ik}\}, \mathbf{r} \in \mathbb{R}_+^{N \times 1}}{\text{maximize}} \quad f(\mathbf{r}) \triangleq \sum_{i \in \mathcal{N}} w_i r_i \quad (4a)$$

$$\text{subject to} \quad \log(1 + \gamma_i(\{\mathbf{v}_{ik}\})) \geq r_i, \forall i \in \mathcal{N} \quad (4b)$$

$$\sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K} \quad (4c)$$

Problem (4) is the epigraph representation of (3), thus the two problems are equal in the sense of optimal solutions.

The following properties make (4) suitable for applying BRnB. First, since $w_i \geq 0$, objective function $f(\mathbf{r})$ is monotonically increasing with respect to \mathbf{r} , i.e., $f(\mathbf{r}') \leq f(\mathbf{r})$ for all $\mathbf{r}' \leq \mathbf{r}$ where the inequality is understood element wise. Second, let $\check{r}_i = 0$ and $\hat{r}_i = \log(1 +$

$\frac{\sum_{k \in \mathcal{K}} P_k \mathbf{h}_{ik} \mathbf{h}_{ik}^H}{\sigma_i^2})$, $\forall i \in \mathcal{N}$, then the box $[\tilde{\mathbf{r}}, \hat{\mathbf{r}}]$ contains all feasible points \mathbf{r} in (4). Third, the set $\mathcal{S} \triangleq \{\mathbf{r} \in \mathbb{R}_+^{N \times 1}, (4b), (4c)\}$ is normal compact in $[\tilde{\mathbf{r}}, \hat{\mathbf{r}}]$, i.e., if $\mathbf{r}' \in \mathcal{S}$, then $[\tilde{\mathbf{r}}, \mathbf{r}'] \subset \mathcal{S}$.

A BRnB monotonic algorithm is an iterative procedure comprising of three main steps called branching, reduction and bounding. In what follows, we present the details of these steps customized to (4).

Let us consider iteration t and define some notations. Let lb_{best} and \mathbf{r}_{best} denote the current best lower bound and the feasible point achieving lb_{best} , respectively. Let \mathcal{Q} be the set of candidate boxes. Let $\text{ub}(R)$ denote an upper bound on $f(\mathbf{r})$ for feasible \mathbf{r} in box R (i.e. $\mathbf{r} \in \mathcal{S} \cap R$). The way to calculate $\text{ub}(R)$ is presented in the bounding step.

Branching

In this step, a box in \mathcal{Q} is picked and then divided into two smaller boxes. Specifically, let $R^{(t)} = [\underline{\mathbf{r}}, \bar{\mathbf{r}}]$ denote the chosen box which has the largest upper bound compared to those in $\mathcal{Q}^{(t)}$, i.e.,

$$R^{(t)} = \underset{R \in \mathcal{Q}}{\text{argmax}} \text{ub}(R). \quad (5)$$

Box $R^{(t)}$ is divided into two boxes $R_1^{(t)} = [\underline{\mathbf{r}}_1, \bar{\mathbf{r}}_1]$ and $R_2^{(t)} = [\underline{\mathbf{r}}_2, \bar{\mathbf{r}}_2]$, $R_1^{(t)} \cup R_2^{(t)} = R^{(t)}$, which are determined as

$$\underline{\mathbf{r}}_1 = \underline{\mathbf{r}} + 0.5(\bar{r}_i - \underline{r}_i)\mathbf{e}_i, \bar{\mathbf{r}}_1 = \bar{\mathbf{r}}, \underline{\mathbf{r}}_2 = \underline{\mathbf{r}}, \bar{\mathbf{r}}_2 = \bar{\mathbf{r}} - 0.5(\bar{r}_i - \underline{r}_i)\mathbf{e}_i \quad (6)$$

where $i \in \mathcal{N}$. The upper bound of the new boxes are $\text{ub}(R_1^{(t)}) = \text{ub}(R)$, and $\text{ub}(R_2^{(t)}) = \min\{\text{ub}(R), f(\bar{\mathbf{r}}_2)\}$.

Remark 1. [Branching with weighted edges] Commonly, the longest edge of $R^{(t)}$ is selected to be branched, i.e., $i = \underset{i' \in \mathcal{N}}{\text{argmax}} (\bar{r}_{i'} - \underline{r}_{i'})$ [45], [46]. For the WSR problem (4), we propose the rule of choosing the longest weighted edge for branching, i.e., $i = \underset{i' \in \mathcal{N}}{\text{argmax}} w_{i'}(\bar{r}_{i'} - \underline{r}_{i'})$. The proposed rule is numerically shown to accelerate the convergence (cf. Fig. 3).

Reduction

The reduction step is to remove the parts of a box which is guaranteed not to contain an optimal solution. Let us consider the box $R = [\underline{\mathbf{r}}, \bar{\mathbf{r}}]$ which has $\text{ub}(R) \geq \text{lb}_{\text{best}}$. The reduction of R is denoted by $\text{red}(R) = [\underline{\mathbf{r}}', \bar{\mathbf{r}}']$, where $\underline{\mathbf{r}} \leq \underline{\mathbf{r}}'$ and $\bar{\mathbf{r}} \geq \bar{\mathbf{r}}'$ express as

$$\underline{\mathbf{r}}' = \bar{\mathbf{r}} - \sum_{i=1}^N \beta_i (\bar{r}_i - \underline{r}_i) \mathbf{e}_i, \bar{\mathbf{r}}' = \underline{\mathbf{r}} + \sum_{i=1}^N \alpha_i (\bar{r}_i - \underline{r}_i) \mathbf{e}_i \quad (7)$$

where

$$\begin{aligned}\beta_i &= \operatorname{argmax}\{\beta | \beta \in [0, 1], f(\bar{\mathbf{r}} - \beta(\bar{r}_i - \underline{r}_i)\mathbf{e}_i) \geq \text{lb}_{\text{best}}\} \\ \alpha_i &= \operatorname{argmax}\{\alpha | \alpha \in [0, 1], f(\underline{\mathbf{r}}' + \alpha(\bar{r}_i - \underline{r}_i')\mathbf{e}_i) \leq \text{ub}(R)\}.\end{aligned}\quad (8)$$

It is guaranteed that any optimal solution contained in R is in $\text{red}(R)$. Indeed, with the $\{\alpha_i\}$ and $\{\beta_i\}$ in (8), we have $f(\underline{\mathbf{r}}') \leq \text{lb}_{\text{best}}$ and $f(\bar{\mathbf{r}}') \geq \text{ub}(R)$. This means only the points $\mathbf{r} \in R$ with $f(\mathbf{r}) < \text{lb}_{\text{best}}$ or $f(\mathbf{r}) > \text{ub}(R)$ are removed. In fact, (8) can be rewritten in the following closed-form expressions

$$\beta_i = \min\left\{1, \frac{f(\bar{\mathbf{r}}) - \text{lb}_{\text{best}}}{w_i(\bar{r}_i - \underline{r}_i)}\right\}, \quad \alpha_i = \min\left\{1, \frac{\text{ub}(R) - f(\underline{\mathbf{r}}')}{w_i(\bar{r}_i - \underline{r}_i')}\right\}.$$

Bounding

This operation is to improve lb_{best} and the upper bound of a box, from which the boxes containing no feasible point whose objective value is larger than the current lb_{best} are removed. Let us consider the box $R = [\underline{\mathbf{r}}, \bar{\mathbf{r}}]$ where $\underline{\mathbf{r}}$ is feasible (otherwise, R does not contain any feasible point, and thus should be removed). Let $\phi = \frac{\bar{\mathbf{r}} - \underline{\mathbf{r}}}{\|\bar{\mathbf{r}} - \underline{\mathbf{r}}\|_2}$, $\delta_{\text{low}} = \operatorname{argmax}\{\delta | (\underline{\mathbf{r}} + \delta\phi) \in \mathcal{S}\}$, and $\delta_{\text{up}} = \operatorname{argmin}\{\delta | (\underline{\mathbf{r}} + \delta\phi) \in [\bar{\mathbf{r}}, \hat{\mathbf{r}}] \setminus \mathcal{S}\}$. Clearly, the best objective value achieved by the feasible points in R lies in the segment $[f(\underline{\mathbf{r}} + \delta_{\text{low}}\phi), f(\underline{\mathbf{r}} + \delta_{\text{up}}\phi)]$. Thus, we can update lb_{best} and $\text{ub}(R)$ as: $\text{lb}_{\text{best}} = \max\{f(\underline{\mathbf{r}} + \delta_{\text{low}}\phi), \text{lb}_{\text{best}}\}$ and $\text{ub}(R) = \min\{\max\{f(\bar{\mathbf{r}} - (\bar{r}_i - \underline{r}_i - \delta_{\text{up}}\phi_i)\mathbf{e}_i | i \in \mathcal{N}\}, \text{ub}(R)\}$. The values of δ_{low} and δ_{up} can be determined by the bisection algorithm over the interval $[0, \|\bar{\mathbf{r}} - \underline{\mathbf{r}}\|_2]$.

Checking Feasibility: From the above discussions, it becomes apparent that checking whether a given point is feasible or not plays the key role in bounding. Given \mathbf{r} , the feasibility problem is given by

$$\tilde{\mathcal{S}}(\mathbf{r}) = \text{find } \{\mathbf{v}_{ik}\} \quad (9a)$$

$$\text{subject to } \frac{\sum_{k \in \mathcal{K}} |\mathbf{h}_{ik} \mathbf{v}_{ik}|^2}{\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2} \geq \tilde{r}_i, \forall i \in \mathcal{N} \quad (9b)$$

$$\sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K} \quad (9c)$$

where $\tilde{r}_i = \exp(r_i) - 1$. The feasible set in (9) is nonconvex due to (9b). Different from the coherent JT, using the trick of a phase rotation here does not lead to a tractable formulation due to the sum of quadratic functions at numerator in (9b) [45]. However, problem (9) can be solved exactly using semidefinite relaxation (SDR). In particular, let us write the SDR of (9) as

$$\tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r}) = \text{find } \{\mathbf{V}_{ik}\} \quad (10a)$$

$$\text{subject to } \sum_{k \in \mathcal{K}} \mathbf{h}_{ik} \mathbf{V}_{ik} \mathbf{h}_{ik}^H \geq \tilde{r}_i \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} \mathbf{h}_{ik} \mathbf{V}_{jk} \mathbf{h}_{ik}^H + \tilde{r}_i \sigma_i^2, \forall i \in \mathcal{N} \quad (10b)$$

$$\sum_{i \in \mathcal{N}} \text{tr}(\mathbf{V}_{ik}) \leq P_k, \forall k \in \mathcal{K}, \mathbf{V}_{ik} \in \mathbb{S}_+^{M_k}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}. \quad (10c)$$

The relationship between (9) and (10) in terms of feasibility is stated in the following lemma.

Lemma 1. *For a given point \mathbf{r} , the set $\tilde{\mathcal{S}}(\mathbf{r})$ is nonempty if and only if $\tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r})$ is nonempty.*

Proof: The proof is given in Appendix I-A. ■

The result in Lemma 1 means that the feasibility of some point \mathbf{r} can be justified via (10).

The Globally Optimal Algorithm

The proposed BRnB algorithm for solving (3) is outlined in Algorithm 1. Line 3 is branching, line 4 is reduction, and lines 5-7 are bounding, as explained above. At the initial stage (line 1), we can randomly generate $\{\mathbf{v}_{ik}\}$ such that (3b) is satisfied, and then determine \mathbf{r}_{best} by letting (4b) hold with equality. Notation UB denotes the current largest upper bound (of the boxes in \mathcal{Q}). Removing boxes that do not contain any optimal solution is shown in line 8. The stopping criterion in line 10 ensures that the output lb_{best} is not lower than $100(1 - \epsilon)\%$ of the optimality. To determine beamforming vectors $\{\mathbf{v}_{ik}^*\}$ achieving \mathbf{r}_{best} , i.e., line 11, we first solve problem

$$\underset{\{\mathbf{V}_{ik}\}}{\text{minimize}} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} \text{tr}(\mathbf{V}_{ik}) \text{ subject to } \{(10b), (10c)\} \quad (11)$$

and denote the obtained solution by $\{\mathbf{V}_{ik}'\}$. If $\text{rank}(\mathbf{V}_{ik}') \leq 1$, we extract \mathbf{v}_{ik}^* via the eigenvalue decomposition of \mathbf{V}_{ik}' . Otherwise, \mathbf{v}_{ik}^* can be found as the solution to the following CQP derived from (32) (see Appendix I-A)

$$\underset{\|\mathbf{v}\|_2^2 \leq \text{tr}(\mathbf{V}_{ik}')}{\text{maximize}} \text{Re}\{\mathbf{h}_{ik} \mathbf{v}\} \text{ subject to } \mathbf{v}^H \mathbf{h}_{jk}^H \mathbf{h}_{jk} \mathbf{v} \leq \mathbf{h}_{jk} \mathbf{V}_{ik}' \mathbf{h}_{jk}^H, \forall j \neq i. \quad (12)$$

We now discuss the optimality of Algorithm 1. In particular, let us denote by f_{opt} the optimal objective value. Then we have the following lemma.

Lemma 2. *Given any $\epsilon > 0$, Algorithm 1 guarantees to achieve $\text{UB} - \text{lb}_{\text{best}} < \epsilon$ where $f_{\text{opt}} \in [\text{lb}_{\text{best}}, \text{UB}]$ in a finite number of iterations.*

Proof: A proof is provided in Appendix I-B. ■

Since \mathbf{r}_{best} is feasible, the lemma means that we can find an ϵ -approximate optimal solution, i.e. $f_{\text{opt}} - f(\mathbf{r}_{\text{best}}) < \epsilon$, for any $\epsilon > 0$ after a finite number of iterations.

Algorithm 1 A Branch Reduce and Bound Algorithm to Globally Solve (3)

- 1: **Initialization:** set $t := 1$, $\mathcal{Q} := [\hat{\mathbf{r}}, \hat{\mathbf{r}}]$, $\text{UB} := f(\hat{\mathbf{r}})$. Set initial \mathbf{r}_{best} and $\text{lb}_{\text{best}} = f(\mathbf{r}_{\text{best}})$.
Given accuracy parameter ϵ .
 - 2: **repeat**
 - 3: Select $R^{(t)}$ as (5), then branch $R^{(t)}$ into $R_1^{(t)}$ and $R_2^{(t)}$ as (6). $\mathcal{Q} := \mathcal{Q} \setminus R^{(t)}$.
 - 4: **if** $\text{ub}(R_m^{(t)}) \geq \text{lb}_{\text{best}}$ where $m \in \{1, 2\}$, **then** determine $\text{red}(R_m^{(t)})$ and update $\mathcal{Q} := \mathcal{Q} \cup \text{red}(R_m^{(t)})$ **end**.
 - 5: **if** $\text{red}(R_1^{(t)}) = [\underline{\mathbf{r}}'_1, \bar{\mathbf{r}}'_1]$ contains feasible points (check feasibility of $\underline{\mathbf{r}}'_1$) **then**
 - 6: determine $\phi = \frac{\bar{\mathbf{r}}'_1 - \underline{\mathbf{r}}'_1}{\|\bar{\mathbf{r}}'_1 - \underline{\mathbf{r}}'_1\|_2}$, δ_{low} and δ_{up} , and update lb_{best} , \mathbf{r}_{best} , and $\text{ub}(\text{red}(R_1^{(t)}))$
 - 7: **else** set $\text{ub}(\text{red}(R_1^{(t)})) = 0$ **end**.
 - 8: Update $\mathcal{Q} := \mathcal{Q} \setminus \{R | \text{ub}(R) < \text{lb}_{\text{best}}\}$.
 - 9: Update $\text{UB} := \max\{\text{ub}(R) | R \in \mathcal{Q}\}$, $t := t + 1$.
 - 10: **until** $\frac{\text{UB} - \text{lb}_{\text{best}}}{\text{lb}_{\text{best}}} \leq \epsilon$
 - 11: Determine beamforming vectors $\{\mathbf{v}_{ik}^*\}$ which achieve \mathbf{r}_{best} .
 - 12: **Output:** \mathbf{r}_{best} , lb_{best} , and $\{\mathbf{v}_{ik}^*\}$.
-

The computational complexity of each iteration in Algorithm 1 is mainly incurred by solving feasibility problems (10) at the bounding process. More explicitly, problem (10) contains $2N \sum_k M_k^2$ real variables, $(N + K)$ constraints in size 1, and N constraints in size $\sum_k 2M_k$ for each k . So, the *worst-case* of computational cost for solving (10) is $\mathcal{O}(\sqrt{K + N(1 + 2 \sum_k M_k)} 4N^2(\sum_k M_k^2)^2(K + N(1 + 4 \sum_k M_k^2)))$ [47]. The number of problems needed to be solved depends on the bisection accuracy, denoted by ϵ_{bi} , of determining δ_{low} and δ_{up} , which is $\mathcal{O}(\log_2(1/\epsilon_{\text{bi}}))$.

IV. EFFICIENT SOLUTION TO (3): AN INAP-BASED ALGORITHM

The globally optimal algorithm presented in the previous section comes at a high computational cost, and might be unsuitable for a real-time practical implementation. In this section, we present a fast converging and low-complexity solution to (3) based on the InAp framework, which was inspired by our earlier work in [44]. To do so, we first transform (3) into an equivalent form where convexity is easily justified as follows

$$\underset{\{\mathbf{v}_{ik}\}, \{\mu_i\}}{\text{maximize}} \sum_{i \in \mathcal{N}} w_i \log(1 + \mu_i) \quad (13a)$$

$$\text{subject to } \frac{\sum_{k \in \mathcal{K}} |\mathbf{h}_{ik} \mathbf{v}_{ik}|^2}{\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2} \geq \mu_i, \forall i \in \mathcal{N}, \quad (13b)$$

$$\sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K} \quad (13c)$$

where $\{\mu_i \geq 0\}$ are newly introduced variables.

Lemma 3. *Let $(\{\mathbf{v}_{ik}^*\}, \{\mu_i^*\})$ be an optimal solution to (13), then $\{\mathbf{v}_{ik}^*\}$ is an optimal solution to (3). Conversely, let $\{\mathbf{v}_{ik}^*\}$ be an optimal solution to (3), then $(\{\mathbf{v}_{ik}^*\}, \{\gamma_i(\{\mathbf{v}_{ik}^*\})\})$ is an optimal solution to (13). Moreover, the two problems have the same optimal objective value.*

Proof: A proof is provided in Appendix I-C. ■

It is clear that the nonconvexity of (4) is due to (13b), which can be equivalently rewritten as

$$(13b) \Leftrightarrow \begin{cases} \sum_{k \in \mathcal{K}} |\mathbf{h}_{ik} \mathbf{v}_{ik}|^2 / u_i \geq \mu_i \\ \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2 \leq u_i \end{cases} \quad (14)$$

where $\{u_i > 0\}$ are slack variables. Note that the quadratic-over-linear function is convex with the involved variables. We introduce $\{u_i\}$ to avoid the function $\sum_{k \in \mathcal{K}} |\mathbf{h}_{ik} \mathbf{v}_{ik}|^2 / \mu_i$, which might lead to numerical problems, since μ_i could be zero. In light of the InAp approach, we use a first order approximation as a convex lower bound to derive an approximate convex problem. More explicitly, let $(\{\mathbf{v}_i^{(t)}\}, \{\mu_i^{(t)}\}, \{u_i^{(t)}\})$ be a feasible point, then the approximate problem is

$$\underset{\{\mathbf{v}_{ik}\}, \{\mu_i\}, \{u_i\}}{\text{maximize}} \quad \sum_{i \in \mathcal{N}} w_i \log(1 + \mu_i) \quad (15a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} (\text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}\} - A_{ik}^{(t)} u_i) \geq \mu_i, \forall i \in \mathcal{N}, \quad (15b)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2 \leq u_i, \forall i \in \mathcal{N}, \quad (15c)$$

$$\sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K} \quad (15d)$$

where $\mathbf{g}_{ik}^{(t)} = (2/u_i^{(t)}) (\mathbf{v}_{ik}^{(t)})^H \mathbf{h}_{ik}^H \mathbf{h}_{ik}$ and $A_{ik}^{(t)} = (|\mathbf{h}_{ik} \mathbf{v}_{ik}^{(t)}| / u_i^{(t)})^2$.

CQP-Based Approximation: Since $w_i, i \in \mathcal{N}$, is generally different, problem (15) containing a mix of exponential and second-order cones is normally treated as a generic convex program. The efficiency of modern convex solvers in solving these generic programs is far less than in solving more standard ones. This motivates us to present a quadratic approximation of the objective to obtain a CQP approximate problem of (15). We achieve this by using a lower bound of the logarithm function given as

$$\log(1 + \mu_i) \geq \log(1 + \mu_i^{(t)}) + 2 - 2\sqrt{(1 + \mu_i^{(t)}) / (1 + \mu_i)}. \quad (16)$$

Algorithm 2 An InAp Algorithm to Efficiently Solve (3)

- 1: **Initialization:** Set small ϵ_{IA} , set $t := 0$, choose initial $(\{\mathbf{v}_i^{(0)}\}, \{\mu_i^{(0)}\}, \{u_i^{(0)}\})$.
 - 2: **repeat**
 - 3: Solve (17), and denote the optimal solution by $(\{\mathbf{v}_i^*\}, \{\mu_i^*\}, \{u_i^*\})$
 - 4: Update $(\{\mathbf{v}_i^{(t+1)}\}, \{\mu_i^{(t+1)}\}, \{u_i^{(t+1)}\}) := (\{\mathbf{v}_i^*\}, \{\mu_i^*\}, \{u_i^*\})$
 - 5: Update $t := t + 1$
 - 6: **until** Convergence on objective value
 - 7: **Output:** $\{\mathbf{v}_i^{(t)}\}$
-

The validity of the bound according to the InAp principles is justified in our recent work [48, Sec. III-E]. Next, by introducing new variables $\{\delta_i\}$ and $\{\pi_i\}$, we arrive at the following CQP approximation

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i \quad (17a)$$

$$\text{subject to } \|[2, (\pi_i - \delta_i)]\|_2 \leq (\pi_i + \delta_i), 1 + \mu_i \geq \delta_i^2, \delta_i \geq 1, \forall i \in \mathcal{N}, \quad (17b)$$

$$(15b), (15c), (15d) \quad (17c)$$

where $\tilde{w}_i^{(t)} = w_i \sqrt{1 + \mu_i^{(t)}}$, $\mathbf{x} \triangleq \{\{\mathbf{v}_{ik}\}, \{\mu_i\}, \{u_i\}, \{\delta_i\}, \{\pi_i\}\}$.

Algorithm and Convergence: The InAp-based iterative procedure is outlined in Algorithm 2, which starts with a random initial point (Step 1). In each iteration, a CQP is solved (Step 3) and the feasible point is updated (Step 13). Successively solving (17) and updating $(\{\mathbf{v}_i^{(t)}\}, \{\mu_i^{(t)}\}, \{u_i^{(t)}\})$ by the optimal solution of (17), we obtain the objective sequence $\{\sum_{i \in \mathcal{N}} w_i \log(1 + \mu_i^{(t)})\}_{t=0}^{\infty}$ which is guaranteed to converge as stated in the following lemma.

Lemma 4. Any sequence $\{\sum_{i \in \mathcal{N}} w_i \log(1 + \mu_i^{(t)})\}_{t=0}^{\infty}$ produced by Algorithm 2 is monotonically increasing and converges.

Proof: A proof is provided in Appendix I-D. ■

Computational Complexity: The computational complexity of the algorithm depends on the arithmetical cost of solving approximate problem (17) in each iteration. Problem (17) includes $2N(2 + M)$ real variables where $M = \sum_{k \in \mathcal{K}} M_k$, $2N$ constraints in size 3, $2N$ constraints in size 1, N constraints in size $K(N - 1) + N + 2$, and one constraint in size $2M_k + 1$ for BS k ,

$\forall k \in \mathcal{K}$. Hence, the worst-case computational cost of using a general interior point method for solving (17) is $\mathcal{O}(\sqrt{1 + 5N + K}4N^2(2 + M)^2(N^2(K + 1) + N(10 - K) + 2M + K))$ [47].

Remark 2. [A first-order solution to (3)] We have introduced a few slack variables to achieve a CQP approximation. This maneuver certainly increases the complexity of the problem and may question the efficacy of the proposed iterative solution. This concern is especially relevant as the feasible set of (3) is expressed as a system of *separable* quadratic convex constraints. Thus it is apparently appealing to approximate the objective of (3) by means of a first-order optimization method. In this way, the resulting program in each iteration has low complexity compared to (17). One of the first-order methods widely used for a nonconvex problem such as (3) is the conditional gradient technique (a.k.a. the Frank-Wolfe method) that have received significant interest recently [49]–[51]. In Appendix II we show how a Frank-Wolfe (FW) type algorithm can be derived to solve (3), where the linear optimization oracle at each iteration admits a closed-form expression. While looking very attractive from a per-iteration cost viewpoint, FW type methods in general converge very slowly, i.e., they need a very large number of iterations to produce a high-accuracy solution. As a consequence, the actual run time of the FW type algorithm is much higher than our proposed solution presented in Algorithm 2. We provide numerical examples to demonstrate this point in Fig. 5.

V. DISTRIBUTED IMPLEMENTATION: A COMBINATION OF INAP AND ADMM

In this section, we develop a decentralized algorithm implementing the InAp-based solution where beamforming vectors are calculated locally at the MEC servers using local CSI. The approach is to use the ADMM to solve the convex approximation subproblem (17), in which (17) is converted to an equivalent transformation so that the ADMM procedure can be distributively implemented.

A. Distributed Formulation

We first rearrange (17) according to the MEC servers as

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i \quad (18a)$$

$$\text{subject to } \|[2, (\pi_i - \delta_i)]\|_2 \leq (\pi_i + \delta_i), 1 + \mu_i \geq \delta_i^2, \delta_i \geq 1, \forall i \in \mathcal{N}, \quad (18b)$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} (\text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}\} - A_{ik}^{(t)} u_i) \geq \mu_i, \forall i \in \mathcal{N}, \quad (18c)$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}_d} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sigma_i^2 \leq u_i, \forall i \in \mathcal{N}, \quad (18d)$$

$$\sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K}. \quad (18e)$$

We observe that (18c) and (18d) are the coupling constraints. Thus, we introduce local and global variables so that these constraints are decoupled among the MEC servers. In particular, we equivalently rewrite (18) into the following form

$$\begin{aligned} & \underset{\substack{\{\mathbf{v}_{ik}\}, \{\mu_i\}, \{u_i\}, \{\delta_i\}, \\ \{\pi_i\}, \{\hat{q}_{ik}\}, \{\tilde{q}_{ik}\}, \{q_{ik}\}, \\ \{\hat{y}_{ik}\}, \{\tilde{y}_{ik}\}, \{y_{ik}\}}}{\text{minimize}} & \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i \end{aligned} \quad (19a)$$

$$\begin{aligned} & \text{subject to} \sum_{k \in \mathcal{K}_1} \text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}\} - \left(\sum_{k \in \mathcal{K}_1} A_{ik}^{(t)} + \sum_{d \in \bar{\mathcal{D}}} \sum_{k \in \mathcal{K}_d} A_{ik}^{(t)} \right) u_i \\ & \quad + \sum_{d \in \bar{\mathcal{D}}} \tilde{y}_{id} \geq \mu_i, \forall i \in \mathcal{N}, \end{aligned} \quad (19b)$$

$$\sum_{k \in \mathcal{K}_d} \text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}\} \geq \hat{y}_{id}, \forall i \in \mathcal{N}, d \in \bar{\mathcal{D}} \quad (19c)$$

$$\sum_{k \in \mathcal{K}_1} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 + \sum_{d \in \bar{\mathcal{D}}} \tilde{q}_{id} + \sigma_i^2 \leq u_i, \forall i \in \mathcal{N}, \quad (19d)$$

$$\sum_{k \in \mathcal{K}_d} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 \leq \hat{q}_{id}, \forall i \in \mathcal{N}, d \in \bar{\mathcal{D}} \quad (19e)$$

$$\hat{q}_{id} = q_{id}, \hat{y}_{id} = y_{id}, \forall i \in \mathcal{N}, \forall d \in \bar{\mathcal{D}} \quad (19f)$$

$$\tilde{q}_{id} = q_{id}, \tilde{y}_{ik} = y_{id}, \forall i \in \mathcal{N}, \forall d \in \bar{\mathcal{D}} \quad (19g)$$

$$(18e), (18b) \quad (19h)$$

where $\bar{\mathcal{D}} = \mathcal{D} \setminus \{1\}$, and $\{\hat{q}_{id}\}$, $\{\tilde{q}_{id}\}$, $\{q_{id}\}$, $\{\hat{y}_{id}\}$, $\{\tilde{y}_{id}\}$ and $\{y_{id}\}$ are newly introduced variables for decomposing (18c) and (18d) into constraints which will be handled locally at the MEC servers; constraints (19f) and (19g) ensure the agreement of the local variables $\{\hat{q}_{id}\}$ and $\{\tilde{q}_{id}\}$, and $\{\hat{y}_{id}\}$ and $\{\tilde{y}_{id}\}$.

Lemma 5. *Let $(\mathbf{x}^*, \{\hat{q}_{id}^*\}, \{\tilde{q}_{id}^*\}, \{q_{id}^*\}, \{\hat{y}_{id}^*\}, \{\tilde{y}_{id}^*\}, \{y_{id}^*\})$ be an optimal solution to (19), then \mathbf{x}^* is an optimal solution to (18). Conversely, let \mathbf{x}^* be an optimal solution to (18), then $(\mathbf{x}^*, \{\hat{q}_{id}^*\}, \{\tilde{q}_{id}^*\}, \{q_{id}^*\}, \{\hat{y}_{id}^*\}, \{\tilde{y}_{id}^*\}, \{y_{id}^*\})$ where $\tilde{q}_{id}^* = q_{id}^* = \hat{q}_{id}^* = \sum_{k \in \mathcal{K}_d} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}^*|^2$ and $\tilde{y}_{id}^* = y_{id}^* = \hat{y}_{id}^* = \sum_{k \in \mathcal{K}_d} \text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}^*\}$, $\forall i \in \mathcal{N}, d \in \bar{\mathcal{D}}$, is an optimal solution to (19).*

Proof: The lemma can be proved by using the similar approach to that of Lemma 3. The detail is omitted for the sake of brevity. \blacksquare

We now rewrite (19) in a more compact form. Without loss of generality, we assume that macro BS is controlled by MEC server 1 and $\bar{\mathcal{D}} = \mathcal{D} \setminus \{1\}$. For notational convenience, let us

denote by $\tilde{\mathbf{x}} \triangleq \{\{\mathbf{v}_{ik}\}_{i \in \mathcal{N}, k \in \mathcal{K}_1}, \{\mu_i\}_{i \in \mathcal{N}}, \{u_i\}_{i \in \mathcal{N}}, \{\delta_i\}_{i \in \mathcal{N}}, \{\pi_i\}_{i \in \mathcal{N}}, \{\tilde{q}_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}, \{\tilde{y}_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}\}$ the local variables at MEC server 1, and define its local feasible set as

$$\tilde{\mathcal{S}} \triangleq \{\tilde{\mathbf{x}} | (19b), (19d), (17b), \sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K}_1\}. \quad (20)$$

Similarly, let us denote by $\hat{\mathbf{x}}_d \triangleq \{\{\mathbf{v}_{ik}\}_{i \in \mathcal{N}, k \in \mathcal{K}_d}, \{\hat{q}_{id}\}_{i \in \mathcal{N}}, \{\hat{y}_{id}\}_{i \in \mathcal{N}}\}$ the local variables at MEC server d , $d \in \bar{\mathcal{D}}$, and define its local feasible set as

$$\hat{\mathcal{S}}_d = \{\hat{\mathbf{x}}_d | \sum_{k \in \mathcal{K}_d} \text{Re}\{\mathbf{g}_{ik}^{(t)} \mathbf{v}_{ik}\} \geq \hat{y}_{id}, \forall i \in \mathcal{N}, \sum_{k \in \mathcal{K}_d} \sum_{j \in \mathcal{N} \setminus \{i\}} |\mathbf{h}_{ik} \mathbf{v}_{jk}|^2 \leq \hat{q}_{id}, \forall i \in \mathcal{N}, \sum_{i \in \mathcal{N}} \mathbf{v}_{ik}^H \mathbf{v}_{ik} \leq P_k, \forall k \in \mathcal{K}_d\}. \quad (21)$$

With these notations, we can rewrite (19) as

$$\begin{aligned} & \underset{\substack{\{\tilde{\mathbf{x}} \in \tilde{\mathcal{S}}\}, \{\hat{\mathbf{x}}_d \in \hat{\mathcal{S}}_d\}_{d \in \bar{\mathcal{D}}} \\ \{q_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}, \{y_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}}}{\text{minimize}} & \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i \end{aligned} \quad (22a)$$

$$\text{subject to } \tilde{\boldsymbol{\pi}} = \tilde{\boldsymbol{\phi}}, \hat{\boldsymbol{\pi}}_d = \hat{\boldsymbol{\phi}}_d, \forall d \in \bar{\mathcal{D}} \quad (22b)$$

where $\tilde{\boldsymbol{\pi}} \triangleq \{\{\tilde{q}_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}, \{\tilde{y}_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}\}$, $\hat{\boldsymbol{\pi}}_d \triangleq \{\{\hat{q}_{id}\}_{i \in \mathcal{N}}, \{\hat{y}_{id}\}_{i \in \mathcal{N}}\}$; $\tilde{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\phi}}_d$ are the rearranged vectors from the same set of global variables $(\{q_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}}, \{y_{id}\}_{i \in \mathcal{N}, d \in \bar{\mathcal{D}}})$.

Now, it can be seen that (22) is in the form of consensus problem which can be solved using the ADMM [35]. We have the augmented Lagrangian function of (22) given by

$$\begin{aligned} \mathfrak{L}(\{\tilde{\mathbf{x}}\}, \{\hat{\mathbf{x}}_d\}, \{q_{id}\}, \{y_{id}\}; \{\boldsymbol{\xi}\}, \{\boldsymbol{\rho}_d\}) = & \left(\boldsymbol{\xi}^T (\tilde{\boldsymbol{\pi}} - \tilde{\boldsymbol{\phi}}) + \frac{m}{2} \|\tilde{\boldsymbol{\pi}} - \tilde{\boldsymbol{\phi}}\|_2^2 + \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i \right) \\ & + \sum_{d \in \bar{\mathcal{D}}} \left(\boldsymbol{\rho}_d^T (\hat{\boldsymbol{\pi}}_d - \hat{\boldsymbol{\phi}}_d) + \frac{m}{2} \|\hat{\boldsymbol{\pi}}_d - \hat{\boldsymbol{\phi}}_d\|_2^2 \right) \end{aligned} \quad (23)$$

where $\boldsymbol{\xi} \in \mathbb{R}^{N(D-1) \times 1}$ and $\{\boldsymbol{\rho}_d\}_{d \in \bar{\mathcal{D}}}, \boldsymbol{\rho}_d \in \mathbb{R}^{N \times 1}$, are the vectors of Lagrangian multipliers; $m > 0$ is the penalty parameter that weighs the violation of the equality constraints. In what follows, we present the variable updates at iteration $(j+1)$ of the ADMM procedure.

B. Variable Updates

1) *Update Local Variables:* Let $\boldsymbol{\xi}^{(j)}$, $\tilde{\boldsymbol{\phi}}^{(j)}$, and $\{\boldsymbol{\rho}_d^{(j)}\}$ be the values obtained at iteration j . MEC server 1 updates its local variables $\tilde{\mathbf{x}}$ by solving the following CQP

$$\underset{\tilde{\mathbf{x}} \in \tilde{\mathcal{S}}}{\text{minimize}} \quad (\boldsymbol{\xi}^{(j)})^T (\tilde{\boldsymbol{\pi}} - \tilde{\boldsymbol{\phi}}^{(j)}) + \frac{m}{2} \|\tilde{\boldsymbol{\pi}} - \tilde{\boldsymbol{\phi}}^{(j)}\|_2^2 + \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i. \quad (24)$$

Also, MEC server d , $d \in \bar{\mathcal{D}}$, updates its local variables $\hat{\mathbf{x}}_d$ by solving the following quadratically constrained quadratic program (QCQP)

$$\underset{\hat{\mathbf{x}}_d \in \hat{\mathcal{S}}_d}{\text{minimize}} \quad (\boldsymbol{\rho}_d^{(j)})^T (\hat{\boldsymbol{\pi}}_d - \hat{\boldsymbol{\phi}}_d^{(j)}) + \frac{m}{2} \|\hat{\boldsymbol{\pi}}_d - \hat{\boldsymbol{\phi}}_d^{(j)}\|_2^2. \quad (25)$$

2) *Update Global Variables:* The global variables $\{q_{id}\}$ and $\{y_{id}\}$ are updated via finding the minimum of the following quadratic function derived from (23)

$$G^{(j)}(\{q_{id}\}, \{y_{id}\}) \triangleq \sum_{d \in \bar{\mathcal{D}}} \sum_{i \in \mathcal{N}} \left([\boldsymbol{\xi}^{(j)}]_{q_{id}} (\tilde{q}_{id}^{(j+1)} - q_{id}) + \frac{m}{2} (\tilde{q}_{id}^{(j+1)} - q_{id})^2 \right. \\ \left. + [\boldsymbol{\xi}^{(j)}]_{y_{id}} (\tilde{y}_{id}^{(j+1)} - y_{id}) + \frac{m}{2} (\tilde{y}_{id}^{(j+1)} - y_{id})^2 + [\boldsymbol{\rho}_d^{(j)}]_{q_{id}} (\hat{q}_{id}^{(j+1)} - q_{id}) + \frac{m}{2} (\hat{q}_{id}^{(j+1)} - q_{id})^2 \right. \\ \left. + [\boldsymbol{\rho}_d^{(j)}]_{y_{id}} (\hat{y}_{id}^{(j+1)} - y_{id}) + \frac{m}{2} (\hat{y}_{id}^{(j+1)} - y_{id})^2 \right) \quad (26)$$

where $[\boldsymbol{\xi}^{(j)}]_{q_{id}}$ is the element in $\boldsymbol{\xi}^{(j)}$ corresponding to constraint $\tilde{q}_{id} = q_{id}$; similar definition is applied to $[\boldsymbol{\xi}^{(j)}]_{y_{id}}$, $[\boldsymbol{\rho}_k^{(j)}]_{q_{id}}$ and $[\boldsymbol{\rho}_d^{(j)}]_{y_{id}}$. The closed-form of the minimizer of (26) is given as

$$q_{id}^{(j+1)} = \frac{([\boldsymbol{\xi}^{(j)}]_{q_{id}} + m\tilde{q}_{id}^{(j+1)}) + ([\boldsymbol{\rho}_d^{(j)}]_{q_{id}} + m\hat{q}_{id}^{(j+1)})}{2m} \quad (27)$$

$$y_{id}^{(j+1)} = \frac{([\boldsymbol{\xi}^{(j)}]_{y_{id}} + m\tilde{y}_{id}^{(j+1)}) + ([\boldsymbol{\rho}_d^{(j)}]_{y_{id}} + m\hat{y}_{id}^{(j+1)})}{2m}. \quad (28)$$

3) *Update Lagrangian Multipliers:* The Lagrangian multipliers are updated as follows

$$\boldsymbol{\xi}^{(j+1)} = \boldsymbol{\xi}^{(j)} + m(\tilde{\boldsymbol{\pi}}^{(j+1)} - \tilde{\boldsymbol{\phi}}^{(j+1)}) \quad (29)$$

$$\boldsymbol{\rho}_d^{(j+1)} = \boldsymbol{\rho}_d^{(j)} + m(\hat{\boldsymbol{\pi}}_d^{(j+1)} - \hat{\boldsymbol{\phi}}_d^{(j+1)}) \quad (30)$$

where $\boldsymbol{\xi}^{(j+1)}$ can be determined at MEC server 1 while $\boldsymbol{\rho}_d^{(j+1)}$ is determined at MEC d , $d \in \bar{\mathcal{D}}$.

C. The Distributed Algorithm

We summarize the proposed distributed algorithm in Algorithm 3. It includes two stages: the inner stage is the ADMM procedure solving InAp subproblems, and the outer stage is the InAp feasible point update using the values obtained at the inner stage (Step 13). The values obtained in the last iteration of the ADMM at InAp iteration t are used for initializing ADMM procedure at InAp iteration $t + 1$ (Step 14). The initial values for the algorithm (Step 1) will be specified in Section VI.

1) *Exchanged Signals:* We now discuss the signaling exchanged between MEC servers for implementing Algorithm 3. In each ADMM iteration (inner stage), MEC server 1 acquires the two scalars $\frac{[\boldsymbol{\rho}_d^{(j)}]_{q_{id}} + m\hat{q}_{id}^{(j+1)}}{2m}$ and $\frac{[\boldsymbol{\rho}_d^{(j)}]_{y_{id}} + m\hat{y}_{id}^{(j+1)}}{2m}$ from MEC server d , $d \in \bar{\mathcal{D}}$, to update the global variables $q_{id}^{(j+1)}$ and $y_{id}^{(j+1)}$. In order to update multipliers $\boldsymbol{\rho}_d^{(j+1)}$, $d \in \bar{\mathcal{D}}$, MEC server d needs the global parameters $q_{id}^{(j+1)}$ and $y_{id}^{(j+1)}$ from MEC server 1. For the outer stage (InAp iteration), after the inner stage converges, MEC server d , $d \in \bar{\mathcal{D}}$, needs $\{u_i^{(t)}\}$ from MEC server 1 to update

Algorithm 3 Decentralized Procedure Solving (3)

- 1: **Initialization:** Set small ϵ_{IA} , $t := 0$ and $j := 0$, choose initial values for $(\{\mathbf{v}_{ik}^{(0)}\}, \{\mu_i^{(0)}\}, \{u_i^{(0)}\})$ and $(\{q_{id}^{(0)}\}, \{y_{id}^{(0)}\}; \boldsymbol{\xi}^{(0)}, \{\boldsymbol{\rho}_d^{(0)}\})$.
 - 2: **repeat** {Outer stage (SCA procedure)}
 - 3: MEC server d , $d \in \bar{\mathcal{D}}$, receives $\{u_i^{(t)}\}$ from MEC server 1 to form $\sum_{k \in \mathcal{K}_d} A_{ik}^{(t)}$ and $\mathbf{g}_{ik}^{(t)}$.
 - 4: MEC server 1 receives scalar $\sum_{k \in \mathcal{K}_d} A_{ik}^{(t)}$ from MEC server d , $d \in \bar{\mathcal{D}}$, to form $\sum_{d \in \bar{\mathcal{D}}} \sum_{k \in \mathcal{K}_d} A_{ik}^{(t)}$.
 - 5: **repeat** {Inner stage (ADMM procedure)}
 - 6: MEC server 1 updates $\tilde{\mathbf{x}}^{(j+1)}$ by solving (24); MEC server d , $d \in \bar{\mathcal{D}}$, updates $\hat{\mathbf{x}}_d^{(j+1)}$ by solving (25).
 - 7: MEC server 1 receives scalars $\frac{([\boldsymbol{\rho}_d^{(j)}]_{q_{id}} + m\hat{q}_{id}^{(j+1)})}{2m}$ and $\frac{([\boldsymbol{\rho}_d^{(j)}]_{y_{id}} + m\hat{y}_{id}^{(j+1)})}{2m}$ from MEC server d , $d \in \bar{\mathcal{D}}$, then updates global variables $q_{id}^{(j+1)}$ and $y_{id}^{(j+1)}$ using (27) and (28).
 - 8: MEC server 1 updates $\boldsymbol{\xi}^{(j+1)}$ by (29); MEC server d , $d \in \bar{\mathcal{D}}$, receives $q_{id}^{(j+1)}$ and $y_{id}^{(j+1)}$ from MEC server 1, then updates $\boldsymbol{\rho}_d^{(j+1)}$ using (30).
 - 9: $j := j + 1$.
 - 10: **until** ADMM convergence
 - 11: Obtain $(\{\mathbf{v}_{ik}^*\}, \{\mu_i^*\}, \{u_i^*\}, \{q_{id}^*\}, \{y_{id}^*\}; \boldsymbol{\xi}^*, \{\boldsymbol{\rho}_d^*\})$, the solution from the ADMM procedure.
 - 12: Update $t := t + 1$, $j := 0$
 - 13: Update $(\{\mathbf{v}_{ik}^{(t)}\}, \{\mu_i^{(t)}\}, \{u_i^{(t)}\}) := (\{\mathbf{v}_{ik}^*\}, \{\mu_i^*\}, \{u_i^*\})$
 - 14: Update $(\{q_{id}^{(0)}\}, \{y_{id}^{(0)}\}; \boldsymbol{\xi}^{(0)}, \{\boldsymbol{\rho}_d^{(0)}\}) := (\{q_{id}^*\}, \{y_{id}^*\}; \boldsymbol{\xi}^*, \{\boldsymbol{\rho}_d^*\})$
 - 15: **until** $\sum_{i \in \mathcal{N}} w_i (\log(1 + \mu_i^{(t+1)}) - \log(1 + \mu_i^{(t)})) \leq \epsilon_{\text{IA}}$
-

$\sum_{k \in \mathcal{K}_d} A_{ik}^{(t)}$ and $\mathbf{g}_{ik}^{(t)}$; and MEC server 1 needs scalar $\sum_{k \in \mathcal{K}_d} A_{ik}^{(t)}$ from MEC server d , $d \in \bar{\mathcal{D}}$. From these, the exchanged signal overhead depends on the numbers of MEC servers and users, and is independent from the number of BSs or the number of transmit antenna.

2) *Convergence of Algorithm 3:* The convergence of Algorithm 3 depends on that of the outer and inner stages. As discussed in Section IV, the outer stage procedure converges when the convex approximate problems are solved optimally. This is achieved by the inner stage procedure as stated in the following lemma.

Lemma 6. *The ADMM procedure of Algorithm 3 guarantees to output a solution which achieves the optimal objective value of (19).*

Proof: A proof is provided in Appendix I-E. ■

Commonly, the convergence of the ADMM procedure is observed via the primal and dual residuals [35, Section 3.3.1]. Specifically, let us define local primal and dual residual vectors as

$$\boldsymbol{\varepsilon}_d^{\text{pri}} \triangleq \begin{cases} \tilde{\boldsymbol{\pi}}^{(j)} - \tilde{\boldsymbol{\phi}}^{(j)} & \text{for } d = 1 \\ \hat{\boldsymbol{\pi}}_d^{(j)} - \hat{\boldsymbol{\phi}}_d^{(j)} & \text{for } d \in \bar{\mathcal{D}} \end{cases} \quad \text{and} \quad \boldsymbol{\varepsilon}_d^{\text{dua}} \triangleq \begin{cases} m(\tilde{\boldsymbol{\phi}}^{(j)} - \tilde{\boldsymbol{\phi}}^{(j-1)}) & \text{for } d = 1 \\ m(\hat{\boldsymbol{\phi}}_d^{(j)} - \hat{\boldsymbol{\phi}}_d^{(j-1)}) & \text{for } d \in \bar{\mathcal{D}} \end{cases}$$

respectively. Also, let us define the local relative tolerances as

$$\bar{\epsilon}_d^{\text{pri}} \triangleq \begin{cases} \bar{\epsilon} \max(\|\tilde{\boldsymbol{\pi}}^{(j)}\|_2, \|\tilde{\boldsymbol{\phi}}^{(j)}\|_2) & \text{for } d = 1 \\ \bar{\epsilon} \max(\|\hat{\boldsymbol{\pi}}_d^{(j)}\|_2, \|\hat{\boldsymbol{\phi}}_d^{(j)}\|_2) & \text{for } d \in \bar{\mathcal{D}} \end{cases} \quad \text{and} \quad \bar{\epsilon}_d^{\text{dua}} \triangleq \begin{cases} \bar{\epsilon} \|\boldsymbol{\xi}^{(j)}\|_2 & \text{for } d = 1 \\ \bar{\epsilon} \|\boldsymbol{\rho}_d^{(j)}\|_2 & \text{for } d \in \bar{\mathcal{D}} \end{cases}$$

where $\bar{\epsilon} > 0$. Then the ADMM procedure terminates when $\|\boldsymbol{\varepsilon}_d^{\text{pri}}\|_2 \leq \bar{\epsilon}_d^{\text{pri}}$ and $\|\boldsymbol{\varepsilon}_d^{\text{dua}}\|_2 \leq \bar{\epsilon}_d^{\text{dua}}$ for all $d \in \mathcal{D}$. This means, each MEC server check its own stopping conditions using its local information, then notifies the others when the stopping criteria are met. The procedure stops when all MEC servers notify the termination.

Remark 3. [Varying penalty parameter] In some cases, varying penalty parameter might help to improve the convergence of the ADMM procedure compared to fixed penalty parameter [35]. A common approach of tuning penalty parameter is using the residual balancing scheme [35]. For decentralization, we can apply the distributed version of the scheme proposed in [52]. In particular, let us denote by $m_d^{(j)}$ the penalty parameter locally used at MEC server d at iteration j , which are updated as

$$m_d^{(j+1)} = \begin{cases} m_d^{(j)} \tau & \text{for } \|\boldsymbol{\varepsilon}_d^{\text{pri}}\|_2 / \bar{\epsilon}_d^{\text{pri}} > \beta \|\boldsymbol{\varepsilon}_d^{\text{dua}}\|_2 / \bar{\epsilon}_d^{\text{dua}} \\ m_d^{(j)} / \tau & \text{for } \|\boldsymbol{\varepsilon}_d^{\text{dua}}\|_2 / \bar{\epsilon}_d^{\text{dua}} > \beta \|\boldsymbol{\varepsilon}_d^{\text{pri}}\|_2 / \bar{\epsilon}_d^{\text{pri}} \\ m_d^{(j)} & \text{otherwise} \end{cases} \quad (31)$$

where $\tau > 1$ and $\beta > 1$ are parameters. To guarantee convergence, all $\{m_d^{(j)}\}_d$ are fixed to a predefined value after a number of ADMM iterations. The approach might help to reach the stopping criteria faster (numerical examples are provided in Fig. 8).

Remark 4. [Maximum number of ADMM iterations] One of the heuristic approaches reducing the total number of ADMM iteration is to set the maximum number of ADMM iterations at each InAp iteration rather than waiting until convergence to stop [33]. We numerically observe that this method also has potential of accelerating Algorithm 3. Numerical examples for this method are shown in Fig. 9.

3) *Computational Complexity at the MEC Servers:* The arithmetical cost at MEC server 1 mainly comes from finding the solution to CQP (24) in each ADMM iteration. Subproblem (24) contains $2N(1 + D + \sum_{k \in \mathcal{K}_1} M_k)$ real variables, $2N$ constraints in size 3, $2N$ constraints in size 1, N constraints in size $|\mathcal{K}_1|(N - 1) + 2$, and one constraint in size $2M_k + 1$ for each $k \in \mathcal{K}_1$. Hence, the worst-case computational cost of using a general interior point method for solving (17) is $\mathcal{O}(\sqrt{5N + |\mathcal{K}_1|} 4N^2(1 + D + \sum_{k \in \mathcal{K}_1} M_k)^2(10N + |\mathcal{K}_1|(N^2 - N + 1) + 2 \sum_{k \in \mathcal{K}_1} M_k))$ [47].

Similarly, in each ADMM iteration, MEC server d , $d \in \bar{\mathcal{D}}$, solves a QCQP (25). The subproblem contains $2N(1 + \sum_{k \in \mathcal{K}_d} M_k)$ real variables, N constraints in size $|\mathcal{K}_d|(N - 1) + 2$, N constraints in size 1, and one constraint in size $2M_k + 1$ for each $k \in \mathcal{K}_d$. So, the worst-case computational cost is $\mathcal{O}(\sqrt{2N + |\mathcal{K}_d|} 4N^2(1 + \sum_{k \in \mathcal{K}_d} M_k)^2(3N + |\mathcal{K}_d|(N^2 - N + 1) + 2 \sum_{k \in \mathcal{K}_d} M_k))$ [47]. We can see that the problem solved at each MEC server has the smaller size compared to the problem solved in the centralized scheme, i.e. (17).

VI. NUMERICAL RESULTS

We now numerically investigate the performance of the noncoherent JT in dense small cell networks. We consider a circular region with a radius of 500m centered at B_1 , and the small cell BSs randomly placed in the annulus with radii 200m and 500m following a uniform distribution. For the channels, both large scale fading (path loss) and small scale fading are taken into account, i.e., the channel vectors are modeled as $\mathbf{h}_{ik} = \sqrt{\ell_{ik}^{-\beta}} \hat{\mathbf{h}}_{ik}$, where $\hat{\mathbf{h}}_{ik} \sim \mathcal{CN}(0, \mathbf{I})$, ℓ_{ik} is the distance in meters, and β is the path loss exponent which is taken as 5. The noise power density is $N_0 = -174$ dBm/Hz. We take the operation bandwidth as 1 MHz. The maximum transmission power at the BSs are $P_1 = 40$ dBm and $P_k = 30$ dBm, $\forall k \in \bar{\mathcal{K}}$. The number of antennas at the BSs are $M_1 = 8$ and $M_k = 2$, $\forall k \in \bar{\mathcal{K}}$. The number of BSs, users, and other parameters are specified in each experiment.

For initial points, we randomly generate beamforming vectors $\{\mathbf{v}_{ik}\}$ so that (3b) is satisfied. All the convex programs in this section are solved by using the solver MOSEK [53] with the modeling toolbox YALMIP [54].

A. Convergence Performance of Algorithms 1 and 2

Fig. 3 shows the numerical examples of convergence of Algorithm 1 with the longest edge branching rule and longest weighted edge branching rule (discussed in Remark 1). For fairness

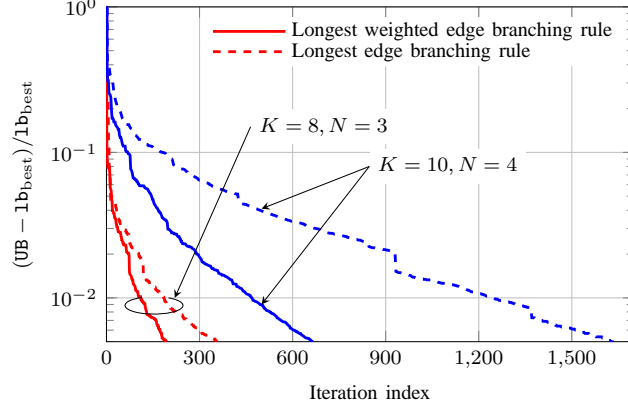
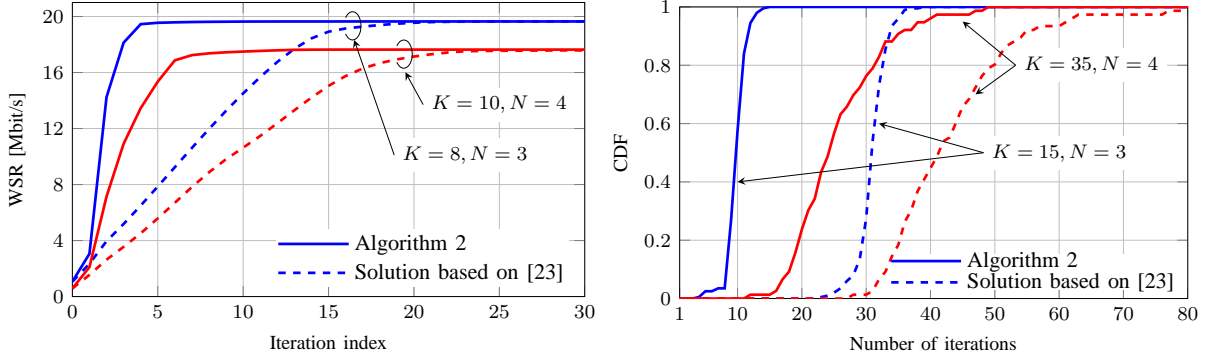


Figure 3. Convergence examples of Algorithm 1 with the longest edge branching rule and longest weighted edge branching rule over two random channel realizations corresponding to the network settings $(K, N) = (8, 3)$ and $(K, N) = (10, 4)$. The weights are taken as $\mathbf{w} = [0.59; 0.31; 0.1]$ for $N = 3$, and $\mathbf{w} = [0.097; 0.519; 0.135; 0.249]$ for $N = 4$.



(a) Convergence behavior of the considered schemes (b) CDF of the number of iteration required to converge.

Figure 4. Convergence speed comparison of Algorithm 2 and the solution modified from the one in [23] with different network settings. The weights are taken as $\mathbf{w} = [0.59; 0.31; 0.1]$ for $N = 3$, and $\mathbf{w} = [0.097; 0.519; 0.135; 0.249]$ for $N = 4$.

comparison, the same initialization of \mathbf{r}_{best} is used for the two branching rules which is determined by letting $r_i = \log(1 + \gamma_i)$ where γ_i is formed from a random feasible point $\{\mathbf{v}_{ik}\}$. In the figure, we plot the accuracy measurement $(\text{UB} - \text{lb}_{\text{best}})/\text{lb}_{\text{best}}$ (which is used as stopping criterion) as the function of the number of iterations. We set the error tolerance parameter as $\epsilon = 0.005$. It is observed that the curves in the figure monotonically go to zero as the number of iterations increases in all cases of channels and branching rules. This is due to the fact that lb_{best} monotonically increases, and the gap between UB and lb_{best} monotonically decreases. Importantly, the results confirm that the longest weighted edge branching rule can accelerate Algorithm 1 compared to the conventional longest edge branching rule.

In Fig. 4, we evaluate the convergence speed of Algorithm 2 compared to the solution modified from the one in [23] with different settings of (K, N) . In particular, Fig. 4(a) plots the convergence behavior of the two schemes over two random channels realizations. Fig. 4(b) plots the cumulative distribution function (CDF) of the required number of iterations to converge. The two schemes stop when the increase in the objective value achieved in the last 3 iterations is less than 10^{-2} . For each channel realization, the same random initial point is used by both schemes for the fairness. We can observe from the figure that the convergence speed of Algorithm 2 is superior in all cases of considered network settings. It is worth mentioning that the two schemes usually, but not always, converge to a same value, and achieve almost the same average performance.

In Fig. 5, we provide an numerical example showing the convergence behavior of the FW solution (see discussion about FW solution in Remark 2 and the detail of the solution in Appendix II) in comparison with the proposed InAp-based solution. For the FW solution, the performance of the diminishing step size rule [50] and the adaptive step size rule [51] are provided. The FW schemes stop when the FW gap is smaller than 1, or the number of iteration exceeds 10^6 . All schemes use the same random starting point. We can observe from the figure that, with the two step size rules, the FW solution requires sufficiently large amount of iterations to reach a good performance.

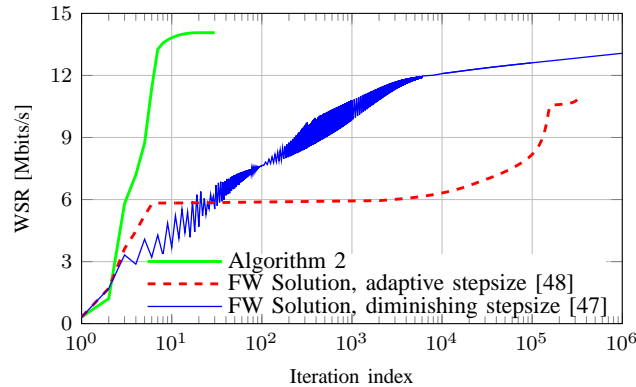


Figure 5. Convergence examples of Algorithm 2 in comparison with FW solution (discussed in Remark 2) with two different step size rules provided in [50] and [51] over a random channel realization corresponding to the network setting $(K, N) = (10, 4)$, $\mathbf{w} = [0.097; 0.519; 0.135; 0.249]$.

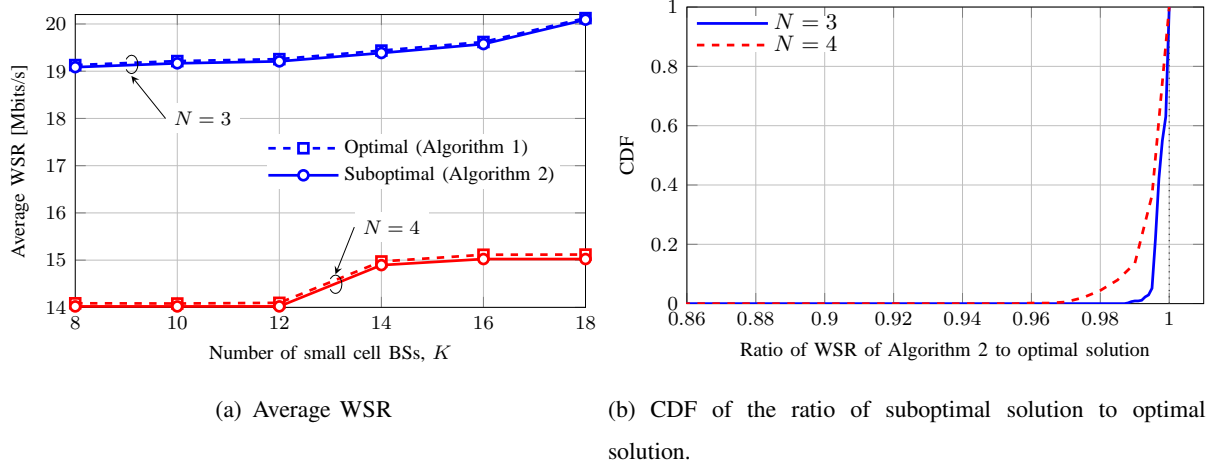


Figure 6. Performance of Algorithm 2 in comparison with optimal solution (Algorithm 1) with different numbers of small cell BSs. The weights are taken as $\mathbf{w} = [0.59; 0.31; 0.1]$ for $N = 3$, and $\mathbf{w} = [0.097; 0.519; 0.135; 0.249]$ for $N = 4$.

B. Performance Comparison between Optimal Solution and Suboptimal InAp-based Solution.

We now evaluate the performance of Algorithm 2 in terms of WSR using the globally optimal solution (Algorithm 3) as the baseline. Specifically, Fig. 6(a) plots the average WSR performance as the function of the number of small cell BSs. Fig. 6(b) provides the CDF of the ratio of WSR of Algorithm 2 to the optimal solution. We can observe from Fig. 6(a) that the average WSR performance of Algorithm 2 is very close to the optimal one. In Fig. 6(b), we see that Algorithm 2 is not worse than 96% of the optimal solution for all channels. The results demonstrate the efficiency of the proposed efficient solution in terms of achieving the design objective.

C. Numerical Results of the Distributed Algorithm

In the next set of the experiments, we examine the performance of Algorithm 3. We consider the decentralized architecture network including 5 MEC servers. Each of them serves the BSs lying in a specific area as shown in Fig. 7. Unless otherwise stated, we set the initial values as $q_{id}^{(0)} = y_{id}^{(0)} = 1$, for all i, d , $\xi^{(0)} = 1$, and $\rho_d^{(0)} = 1$ for all d .

Fig. 8 shows the function $\max(\{||\epsilon_d^{\text{pri}}||_2/\bar{\epsilon}_d^{\text{pri}}\}_d, \{||\epsilon_d^{\text{dua}}||_2/\bar{\epsilon}_d^{\text{dua}}\}_d)$ over the ADMM iterations in the first InAp iteration of a random channel realization. The function is introduced based on the stopping criterion, which is satisfied when the value of the function is smaller than 1. For the adaptive penalty scheme, we take the relative tolerance $\bar{\epsilon}$ as 10^{-3} [35]. Other parameters are taken as $\tau = 2$, $\beta = 5$. The initial values of penalty parameters are set as $m_d = m_0$ for all d ; and after 50 iterations, they are fixed as $m_d = m_0$ for all d [52]. For the fixed penalty parameter

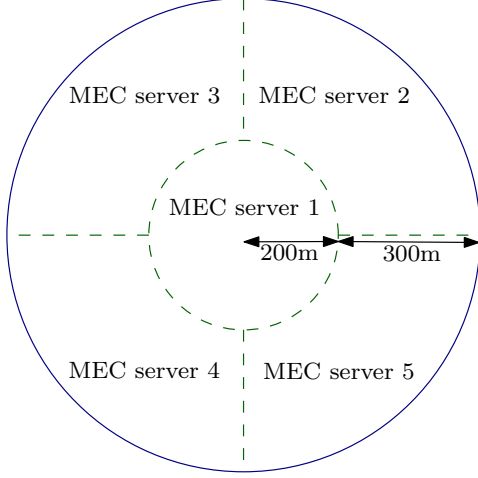


Figure 7. The specified serving areas of the MEC servers considered in the simulations of decentralized networks.

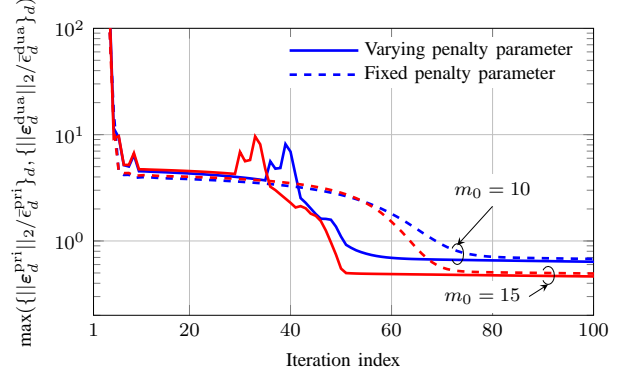


Figure 8. Relative residuals of fixed and varying penalty parameter schemes over ADMM iterations at the first InAp iteration of a random channel realization. The network setting is $(K, N) = (16, 5)$. We take $w_i = 1/N$ for all i .

scheme, the penalty parameter is set as $m = m_0$. Value of m_0 is specified in the figure. We can observe from the figure that, with the same chosen m_0 , adaptive (penalty) scheme can reach the stopping criteria faster.

In Fig. 9, we show the achieved WSR of Algorithm 3 compared to the centralized solution (obtained by Algorithm 2 using the solver) over two random channel realizations. Specifically, Fig. 9(a) plots the achieved WSR as a function of the total number of ADMM iterations, while Fig. 9(b) shows the relative gap between the centralized and distributed solutions. To this end, let us denote by f the WSR at an ADMM iteration which is calculated by using the beamforming vectors obtained at the iteration, and denote by f^* the centralized solution. Then the relative gap is defined as $|f - f^*|/f^*$. We take relative tolerance $\bar{\epsilon}$ as 10^{-3} . In each InAp iteration, the ADMM procedure stops when the termination criterion is met or when the number of ADMM iterations exceeds I_{ADMM} . We take I_{ADMM} as 20 and 70. This consideration is to illustrate the heuristic usage of the ADMM discussed in Remark 4. We can observe that the total number of ADMM iterations reduces remarkably with appropriate value of I_{ADMM} . We note that the WSR over the ADMM iterations is not necessary to be monotonic because the ADMM works on the augmented Lagrangian function. We also note that when the ADMM procedure converges, the sequence of WSR values corresponding to the outer stage is monotonically increasing as Algorithm 2. However, in this figure, since I_{ADMM} is applied, the sequence can be not monotonic since the parameters of the outer stage are updated even when the ADMM has not converged.

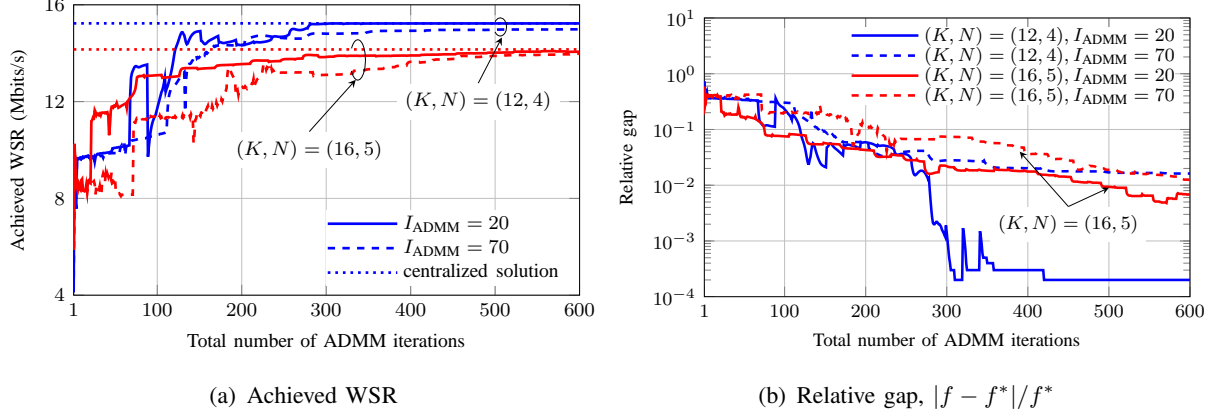


Figure 9. Achieved WSR of Algorithm 3 compared to centralized solution over two random channel realizations corresponding to network settings $(K, N) = (12, 4)$ and $(K, N) = (16, 5)$. We take $\mathbf{w} = [0.097; 0.519; 0.135; 0.249]$ for $N = 4$, and $w_i = 1/N$, for all i , for $N = 5$.

D. Performance Comparison between Different CoMP Strategies

In the final set of experiments, we study the WSR performance of the noncoherent JT in comparison with the other CoMP strategies, i.e., coherent JT, and CB. In CB scheme, each of users is only served by the nearest BS. We provide the WSR performance of the noncoherent JT based on Algorithm 2 due to its low complexity and near-optimal performance. We also provide the noncoherent JT using MRT scheme as a benchmark. The solutions of the coherent JT and CB are derived based on that in [44].

Fig. 10 plots the average WSR performance of the considered schemes as functions of the number of small cell BSs. An interesting result observed from the figure is that CB scheme might fail to exploit the densification gain. Another result is that the coherent JT is naturally superior to the others. Therefore, in the networks where the coherent JT is feasible, this scheme should be deployed to achieve maximum spectral efficiency. However, when the synchronization accuracy is not sufficient, the noncoherent JT with the proposed solution is a promising candidate for dense small cell networks, since it outperforms the MRT scheme and CB, and is capable of exploiting densification gain (the performance increases when K increases).

In Fig. 11, we show the average WSR performance of the considered schemes as the functions of the number of users N for two cases of K . The main result observed from the figure is that the WSR performance of the noncoherent JT with proposed solution, coherent JT and CB increase when N increases. This implies that the three schemes are capable of exploiting the user diversity gain. The results also confirm the observations taken from Fig. 10, i.e., the coherent JT

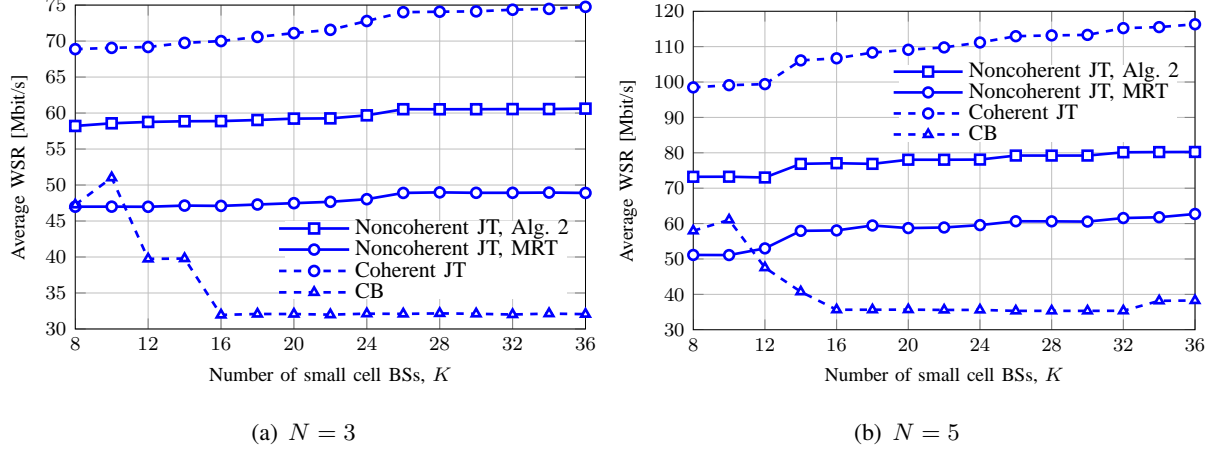


Figure 10. Average WSR performance of the considered noncoherent JT schemes, coherent JT, and CB with different numbers of small cell BSs. The weights are taken as $\mathbf{w} = \mathbf{1}$ for all cases of N .

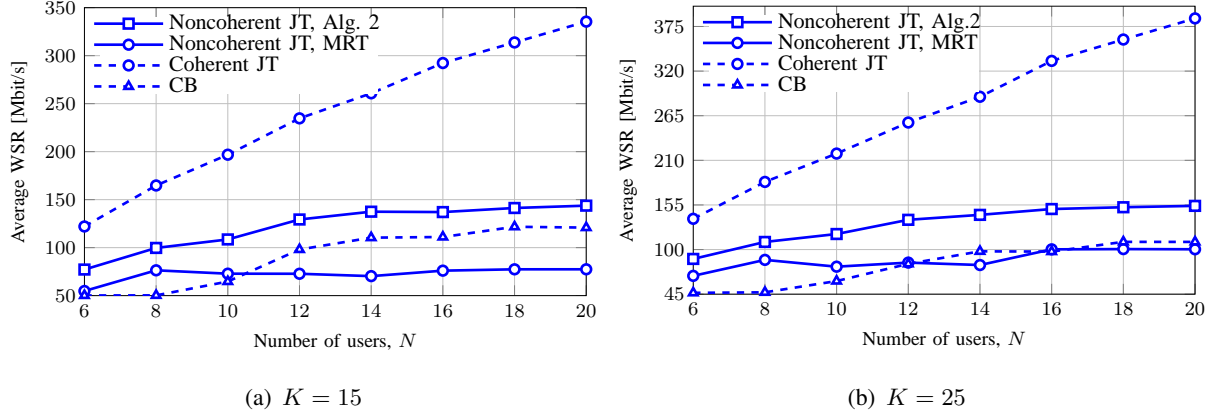


Figure 11. Average WSR performance of the considered noncoherent JT schemes, coherent JT, and CB with different numbers of users. We take K as 15 and 25. The weights are taken as $\mathbf{w} = \mathbf{1}$ for all cases of N .

outperforms the others, and the noncoherent JT with the proposed solution outperforms the MRT scheme and CB in all cases of N . We can see that, when N is large, the CB outperforms the noncoherent JT with the MRT scheme. This might be because the MRT scheme only considers power allocation, and thus it cannot provide good interference management.

VII. CONCLUSION

We have investigated downlink noncoherent JT in dense small cell networks. Particularly, we have considered the problem of designing beamforming vectors at the macro cell and small cell BSs for maximizing WSR. Because the problem is intractable, we have developed a BRnB algorithm to achieve globally optimal solution. In addition, for practical implementation, we

have developed a low-complexity algorithm based on the IA optimization framework, which has been numerically shown to be able to achieve near optimal performance. Moreover, in order to implement the InAp-based solution on decentralized networks using MEC servers, we have provided a distributed algorithm based on the ADMM. The results have revealed that noncoherent JT is capable of exploiting densitification gain and outperforming the CB. It is also feasible to implement the transmission scheme distributively, and it does not require as strict synchronization accuracy as the coherent JT. Thus, noncoherent JT is a promising transmission scheme for dense small cell networks in terms of the WSR performance.

APPENDIX I

PROOF OF LEMMAS

A. Proof of Lemma 1

The proof is based on that of [21, Theorem 1]. The if part is obvious since $\tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r})$ is the SDR of $\tilde{\mathcal{S}}(\mathbf{r})$ achieved by removing rank-one constraints. Thus we focus on the only-if part, i.e., if $\tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r})$ is nonempty, $\tilde{\mathcal{S}}(\mathbf{r})$ is nonempty. If there exists $\mathbf{V}'_{ik} \in \tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r})$ such that $\text{rank}(\mathbf{V}_{ik}) \leq 1, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}$, then $\tilde{\mathcal{S}}(\mathbf{r})$ is nonempty which completes the proof. Now suppose that there exists $\mathbf{V}'_{uv} \in \tilde{\mathcal{S}}_{\text{SD}}(\mathbf{r})$ where $\text{rank}(\mathbf{V}'_{uv}) > 1$ for some (u, v) . Then consider the problem

$$\underset{\mathbf{V}}{\text{maximize}} \quad \mathbf{h}_{uv} \mathbf{V} \mathbf{h}_{uv}^H \quad (32a)$$

$$\text{subject to} \quad \mathbf{h}_{jv} \mathbf{V} \mathbf{h}_{jv}^H \leq \mathbf{h}_{jv} \mathbf{V}'_{uv} \mathbf{h}_{jv}^H, \forall j \neq u \quad (32b)$$

$$\text{tr}(\mathbf{V}) \leq \text{tr}(\mathbf{V}'_{uv}), \mathbf{V} \in \mathbb{S}_+^{M_v}. \quad (32c)$$

It is proved based on a primal-dual analysis that (32) always has a rank-one solution (c.f [55, Appendix III]), which completes the proof.

B. Proof of Lemma 2

The proof is based on the convergence analysis of BRnB in [30], [46]. Particularly, the reduction is valid, i.e., no feasible point in a box having better performance than lb_{best} is lost after the reduction. The bounding guarantees that the upper bound of a box is non-increasing and lb_{best} is non-decreasing. Thus the gap $\text{UB} - \text{lb}_{\text{best}}$ is monotonically decreasing. In addition, the bisection partition in the branching is exhaustive [30]. These lead to $(\text{UB} - \text{lb}_{\text{best}}) \rightarrow 0$ as $t \rightarrow \infty$, i.e., the gap of the bounds uniformly converges to zero. This completes the proof.

C. Proof of Lemma 3

To prove the first statement, we show that: (i) the constraints in (13b) hold with equality at optimality, and (ii) vectors $\{\mathbf{v}_{ik}^*\}$ achieve optimal value of (3). For (i), suppose that (13b) is inactive at the optimality for some i , then there exists $\mu'_i > \mu_i^*$ such that constraint i in (13b) holds with equality. Thus, the point with μ'_i instead of μ_i^* is still feasible and results in a strictly larger objective value, since $\log(1 + \mu'_i) > \log(1 + \mu_i^*)$. This contradicts with the assumption that an optimal solution has been achieved.

For (ii), suppose that there exists a feasible point of (3) denoted by $\{\mathbf{v}'_{ik}\}$ such that $\sum_{i \in \mathcal{N}} w_i \log(1 + \gamma_i(\{\mathbf{v}'_{ik}\})) > \sum_{i \in \mathcal{N}} w_i \log(1 + \mu_i^*)$, then we determine $\mu'_i = \gamma_i(\{\mathbf{v}'_{ik}\})$. Clearly, $(\{\mathbf{v}'_{ik}\}, \{\mu'_i\})$ is a feasible point of (13) which achieves larger objective value than $(\{\mathbf{v}_{ik}^*\}, \{\mu_i^*\})$. Again, this contradicts with the assumption that $(\{\mathbf{v}_{ik}^*\}, \{\mu_i^*\})$ is optimal.

Similarly, for the converse, we can show by contradiction that there exist no feasible points of (13) which can achieve better objective value than $(\{\mathbf{v}_{ik}^*\}, \{\gamma_i(\{\mathbf{v}_{ik}^*\})\})$. Finally, the the final statement follows the fact (i).

D. Proof of Lemma 4

By contradiction, we can justify that at optimality all constraints in (17b) hold with equality, leading to $\sqrt{1 + \mu_i^{(t)}} = 1/\pi_i^{(t)} \forall i, t$. Let us consider iteration t . According to the principles of the InAp, the solution to (17) in iteration t is feasible to the problem in iteration $(t + 1)$, and thus $\sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i^{(t)} \geq \sum_{i \in \mathcal{N}} \tilde{w}_i^{(t)} \pi_i^{(t+1)}$ which is equivalent to

$$\begin{aligned} \sum_{i \in \mathcal{N}} w_i \sqrt{(1 + \mu_i^{(t)})/(1 + \mu_i^{(t)})} &\geq \sum_{i \in \mathcal{N}} w_i \sqrt{(1 + \mu_i^{(t)})/(1 + \mu_i^{(t+1)})} \\ \Leftrightarrow \sum_{i \in \mathcal{N}} w_i \underbrace{(J^{(t)} - 2\sqrt{(1 + \mu_i^{(t)})/(1 + \mu_i^{(t)})})}_{=\log(1 + \mu_i^{(t)})} &\leq \sum_{i \in \mathcal{N}} w_i \underbrace{(J^{(t)} - 2\sqrt{(1 + \mu_i^{(t)})/(1 + \mu_i^{(t+1)})})}_{\leq \log(1 + \mu_i^{(t+1)})} \end{aligned}$$

where $J^{(t)} = \log(1 + \mu_i^{(t)}) + 2$. In addition, problem (13) is upper bounded by a finite value due to power constraints (13c). This completes the proof.

E. Proof of Lemma 6

The proof is based on the convergence analysis of ADMM in [35]. Firstly, we note that the feasible set of (17) is convex and nonempty for all InAp iterations, and so is that of (19). The objective of (17) is bounded. Consequently, sets $\tilde{\mathcal{S}}$ and $\hat{\mathcal{S}}_d$ are nonempty, and the problems (24), (25), and (26) are solvable [35, Assumption 1]. Secondly, we recall that the considered

problem is only constrained by the maximum transmit power at the BSs. Thus, problem (19) is strictly feasible. Consequently, the assumption that the unaugmented Lagrangian (i.e., function (23) with $m = 0$) has a saddle point holds [35, Assumption 2]. With these, the lemma follows the statement in [35, Section 3.2.1].

APPENDIX II

A FIRST-ORDER ALGORITHM SOLVING (3) VIA CONDITIONAL GRADIENT METHOD

In this appendix, we present a first-order solution for solving (3) using Frank-Wolfe's method. To proceed, let us define some notations as $\mathbf{H}_{ik} = \mathbf{h}_{ik}^H \mathbf{h}_{ik}$, $\hat{\mathbf{H}}_{ik} = \text{blkdiag}\{\underbrace{\mathbf{H}_{ik}, \dots, \mathbf{H}_{ik}}_N\}$, $\mathbf{H}_i = \text{blkdiag}\{\hat{\mathbf{H}}_{i1}, \dots, \hat{\mathbf{H}}_{i(k+1)}\}$, $\mathbf{v}_k = [\mathbf{v}_{1k}; \dots; \mathbf{v}_{Nk}]$, $\mathbf{v} = [\mathbf{v}_1; \dots; \mathbf{v}_{k+1}]$, $\hat{\mathbf{G}}_{ik} = \text{blkdiag}\{\underbrace{\mathbf{H}_{ik}, \dots, \mathbf{H}_{ik}}_{i-1}, \mathbf{0}, \underbrace{\mathbf{H}_{ik}, \dots, \mathbf{H}_{ik}}_{N-i}\}$, and $\mathbf{G}_i = \text{blkdiag}\{\hat{\mathbf{G}}_{i1}, \dots, \hat{\mathbf{G}}_{i(k+1)}\}$. Then we rewrite the objective function as

$$g(\mathbf{v}) = \sum_{i \in \mathcal{N}} w_i \log \left(\frac{\mathbf{v}^H \mathbf{H}_i \mathbf{v} + \sigma_i^2}{\mathbf{v}^H \mathbf{G}_i \mathbf{v} + \sigma_i^2} \right)$$

For ease of exposition, we convert $g(\mathbf{v})$ in the real-domain as

$$g(\tilde{\mathbf{v}}) = \sum_{i \in \mathcal{N}} w_i \log \left(\frac{\tilde{\mathbf{v}}^T \bar{\mathbf{H}}_i \tilde{\mathbf{v}} + \sigma_i^2}{\tilde{\mathbf{v}}^T \bar{\mathbf{G}}_i \tilde{\mathbf{v}} + \sigma_i^2} \right)$$

where $\tilde{\mathbf{v}} = [\text{Re}\{\mathbf{v}\}; \text{Im}\{\mathbf{v}\}]$, $\bar{\mathbf{H}}_i = [\text{Re}\{\mathbf{H}_i\}, -\text{Im}\{\mathbf{H}_i\}; \text{Im}\{\mathbf{H}_i\}, \text{Re}\{\mathbf{H}_i\}]$, and $\bar{\mathbf{G}}_i = [\text{Re}\{\mathbf{G}_i\}, -\text{Im}\{\mathbf{G}_i\}; \text{Im}\{\mathbf{G}_i\}, \text{Re}\{\mathbf{G}_i\}]$. Let $\tilde{\mathbf{v}}^{(t)}$ be a feasible point, then the gradient of $g(\tilde{\mathbf{v}})$ at $\tilde{\mathbf{v}}^{(t)}$ is

$$\nabla_{\tilde{\mathbf{v}}} g(\tilde{\mathbf{v}}^{(t)}) = \sum_{i \in \mathcal{N}} w_i \left(\frac{2\bar{\mathbf{H}}_i \tilde{\mathbf{v}}^{(t)}}{(\tilde{\mathbf{v}}^{(t)})^T \bar{\mathbf{H}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2} - \frac{2\bar{\mathbf{G}}_i \tilde{\mathbf{v}}^{(t)}}{(\tilde{\mathbf{v}}^{(t)})^T \bar{\mathbf{G}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2} \right). \quad (33)$$

Then the linear optimization oracle at each iteration is

$$\tilde{\mathbf{v}}^* = \arg\max (\nabla g(\tilde{\mathbf{v}}^{(t)})^T \tilde{\mathbf{v}} | \tilde{\mathbf{v}}_k^T \tilde{\mathbf{v}}_k \leq P_k, \forall k \in \mathcal{K}) \quad (34)$$

where $\tilde{\mathbf{v}}_k = [\text{Re}\{\mathbf{v}_k\}; \text{Im}\{\mathbf{v}_k\}]$. As the objective and the feasible set of the above problem are separable with respect to $\tilde{\mathbf{v}}_k$, it is easy to see that (34) has the following closed-form expression

$$\tilde{\mathbf{v}}_k^* = \sqrt{P_k / (\mathbf{c}_k^T \mathbf{c}_k)} \mathbf{c}_k \quad (35)$$

where $\mathbf{c}_k = \{\nabla g(\tilde{\mathbf{v}}^{(t)})\}_{\tilde{\mathbf{v}}_k}$ is the vector including the elements of $\nabla g(\tilde{\mathbf{v}}^{(t)})$ associated with $\tilde{\mathbf{v}}_k$.

The first-order iterative algorithm for solving (3) is outlined as follows:

Initialization: Set small ϵ_G , $t := 1$, choose initial point $\tilde{\mathbf{v}}^{(1)}$.

repeat

Generate $\nabla g(\tilde{\mathbf{v}}^{(t)})$, then determine $\tilde{\mathbf{v}}^*$ according to (35).

if $\nabla g(\tilde{\mathbf{v}}^{(t)})^\top (\tilde{\mathbf{v}}^* - \tilde{\mathbf{v}}^{(t)}) \leq \epsilon_G$

Stop and return $\tilde{\mathbf{v}}^{(t)}$

else

Choose step size $\alpha^{(t)} \in [0, 1]$, then update $\tilde{\mathbf{v}}^{(t+1)} := (1 - \alpha^{(t)})\tilde{\mathbf{v}}^{(t)} + \alpha^{(t)}\tilde{\mathbf{v}}^*$

$t := t + 1$.

end

In each iteration, the new iterate $\tilde{\mathbf{v}}^{(t+1)}$ is determined by moving the current iterate $\tilde{\mathbf{v}}^{(t)}$ along the dictated direction $\tilde{\mathbf{v}}^* - \tilde{\mathbf{v}}^{(t)}$ with step size $\alpha^{(t)} \in (0, 1]$.

Smoothness of $g(\mathbf{v})$

We now investigate the smoothness of $g(\tilde{\mathbf{v}})$, which is essential for the convergence result of the Frank-Wolfe (FW) based method above to be provided in the next subsection. In particular, the Hessian of $g(\mathbf{v})$ at $\tilde{\mathbf{v}}^{(t)}$ is given by

$$\begin{aligned} \nabla_{\tilde{\mathbf{v}}}^2 g(\tilde{\mathbf{v}}^{(t)}) = \sum_{i \in \mathcal{N}} w_i & \left(\frac{2\bar{\mathbf{H}}_i}{(\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{H}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2} - \frac{4\bar{\mathbf{H}}_i \tilde{\mathbf{v}}^{(t)} (\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{H}}_i^\top}{((\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{H}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2)^2} \right. \\ & \left. - \frac{2\bar{\mathbf{G}}_i}{(\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{G}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2} + \frac{4\bar{\mathbf{G}}_i \tilde{\mathbf{v}}^{(t)} (\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{G}}_i^\top}{((\tilde{\mathbf{v}}^{(t)})^\top \bar{\mathbf{G}}_i \tilde{\mathbf{v}}^{(t)} + \sigma_i^2)^2} \right). \end{aligned} \quad (36)$$

It is trivial to show that $\|\nabla_{\tilde{\mathbf{v}}}^2 g(\tilde{\mathbf{v}}^{(t)})\|_2 \leq \rho_L \triangleq \sum_{i \in \mathcal{N}} w_i \left(\frac{2}{\sigma_i^2} \|\bar{\mathbf{H}}_i\|_2 + \frac{4P_{\text{total}}}{\sigma_i^4} \|\bar{\mathbf{H}}_i\|_2^2 + \frac{2}{\sigma_i^2} \|\bar{\mathbf{G}}_i\|_2 + \frac{4P_{\text{total}}}{\sigma_i^4} \|\bar{\mathbf{G}}_i\|_2^2 \right)$, where $P_{\text{total}} = \sum_k P_k$, which means $g(\tilde{\mathbf{v}})$ is a ρ_L -smooth function [56].

Step Size Rules and Convergence

Note that the FW-type method for nonconvex problems is not monotonically increasing, and thus a proper choice of the step size $\alpha^{(t)}$ at each iteration is critical for the convergence. This issue is relatively open and receiving increasing interest but there are some known step size rules for the conditional gradient method. One is the line-search rule: $\alpha^{(t)} = \arg\max(g((1-\alpha)\tilde{\mathbf{v}}^{(t)} + \tilde{\mathbf{v}}^*) | \alpha^{(t)} \in [0, 1])$ [49], but it is not cost effective here since the one-dimensional search does not admit a closed-form solution. An adaptive step size rule is recently proposed in [51]. The diminishing step size rule $\alpha^{(t)} = t^{-\omega}$, where $\omega \in (0.5, 1]$, can also be used [50].

For the convergence, we recall that the feasible set of problem (3) is convex and compact. In addition, we have shown that $g(\tilde{\mathbf{v}})$ has a finite Lipschitz gradient constant ρ_L . Thus, by the mentioned step size rules, it is guaranteed that the iterative procedure converges to a stationary

point of (3) [49]–[51]. In addition, for the line-search and adaptive step size rules, it is proved that the convergence rate is $\mathcal{O}(1/\sqrt{t})$ [49, Theorem 1], [51, Theorem 1].

REFERENCES

- [1] Q.-D. Vu, L.-N. Tran, and M. Juntti, “Distributed noncoherent transmit beamforming for dense small cell networks,” in *Proc. IEEE ICASSP 2019*, May 2019.
- [2] Ericsson, “Mobility report,” Nov. 2018.
- [3] 3GPP, “Study on scenarios and requirements for next generation access technologies,” TR 38.913, 2017.
- [4] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, “5G ultra-dense cellular networks,” *IEEE Wireless Commun. Mag.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.
- [5] J. Hoydis, M. Kobayashi, and M. Debbah, “Green small-cell networks,” *IEEE Vehicular Technology Magazine*, vol. 6, no. 1, pp. 37–43, March 2011.
- [6] Small Cell Forum, “Small cell siting challenges,” Feb. 2017.
- [7] P. Rost, C. J. Bernardos, A. D. Domenico, M. D. Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wübben, “Cloud technologies for flexible 5G radio access networks,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.
- [8] Ericsson, “Cloud RAN-the benefits of virtualization, centralization and coordination,” Sep. 2015.
- [9] M. Peng, C. Wang, V. Lau, and H. V. Poor, “Fronthaul-constrained cloud radio access networks: insights and challenges,” *IEEE Wireless Commun. Mag.*, vol. 22, no. 2, pp. 152–160, April 2015.
- [10] S. C. Forum, “Edge computing made simple,” Dec. 2017.
- [11] ETSI, “Cloud RAN and MEC: A perfect pairing,” The European Telecommunications Standards Institute, White paper, Feb. 2018.
- [12] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges,” *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, April 2017.
- [13] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.
- [14] Ericsson, “Distributed cloud infrastructure,” 2017.
- [15] ETSI, “Mobile edge computing - a key technology towards 5G,” The European Telecommunications Standards Institute, White paper, Sep. 2015.
- [16] J. Zheng, Y. Wu, N. Zhang, H. Zhou, Y. Cai, and X. Shen, “Optimal power control in ultra-dense small cell networks: A game-theoretic approach,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4139–4150, July 2017.
- [17] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, “Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges,” *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012.
- [18] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, “Multi-cell MIMO cooperative networks: A new look at interference,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec 2010.
- [19] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, “Cloud RAN for Mobile Networks—A Technology Overview,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.
- [20] G. C. Alexandropoulos, P. Ferrand, J. Gorce, and C. B. Papadias, “Advanced coordinated beamforming for the downlink of future LTE cellular networks,” *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 54–60, July 2016.
- [21] E. Björnson, M. Kountouris, and M. Debbah, “Massive MIMO and small cells: Improving energy efficiency by optimal soft-cell coordination,” in *20th International Conference on Telecommunications (ICT)*, May 2013, pp. 1–5.

- [22] J. Li, E. Björnson, T. Svensson, T. Eriksson, and M. Debbah, "Joint precoding and load balancing optimization for energy-efficient heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5810–5822, Oct 2015.
- [23] C. Pan, H. Ren, M. El-kashlan, A. Nallanathan, and L. Hanzo, "The non-coherent ultra-dense C-RAN is capable of outperforming its coherent counterpart at a limited fronthaul capacity," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2549–2560, Nov. 2018.
- [24] T. V. Chien, E. Björnson, and E. G. Larsson, "Joint power allocation and user association optimization for massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6384–6399, Sept 2016.
- [25] J. Proakis, *Digital Communications*, 4th ed., J. Proakis, Ed. McGraw-Hill, 2001.
- [26] Lizhong Zheng and D. N. C. Tse, "Communication on the Grassmann manifold: a geometric approach to the noncoherent multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 359–383, Feb. 2002.
- [27] A. Lapidoth and S. M. Moser, "Capacity bounds via duality with applications to multiple-antenna systems on flat-fading channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2426–2467, Oct 2003.
- [28] M. Kobayashi and G. Caire, "An iterative water-filling algorithm for maximum weighted sum-rate of Gaussian MIMO-BC," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1640–1646, Aug 2006.
- [29] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," vol. 2, no. 1, pp. 57–73, Feb 2008.
- [30] H. Tuy, F. Al-Khayyal, and P. T. Thach, "Monotonic optimization: Branch and cut methods," in *Essays and Surveys in Global Optimization*. Springer, 2005, pp. 39–78.
- [31] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, August 1978.
- [32] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *Journal of Global Optimization*, vol. 47, no. 1, pp. 29–51, 2010.
- [33] K.-G. Nguyen, Q.-D. Vu, M. Juntti, and L.-N. Tran, "Distributed solutions for energy efficiency fairness in multicell MISO downlink," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6232–6247, Sept 2017.
- [34] Q.-D. Vu, L.-N. Tran, M. Juntti, and E.-K. Hong, "Energy-efficient bandwidth and power allocation for multi-homing networks," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1684–1699, April 2015.
- [35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [36] C. Shen, T. Chang, K. Wang, Z. Qiu, and C. Chi, "Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2988–3003, June 2012.
- [37] Z. Xu, C. Yang, G. Li, Y. Liu, and S. Xu, "Energy-efficient CoMP precoding in heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 1005–1017, Feb 2014.
- [38] Q.-D. Vu, L.-N. Tran, R. Farrell, and E.-K. Hong, "Energy-efficient zero-forcing precoding design for small-cell networks," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 790–804, Feb 2016.
- [39] T. X. Tran and D. Pompili, "Dynamic radio cooperation for user-centric Cloud-RAN with computing resource sharing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2379–2393, April 2017.
- [40] T. T. Vu, D. T. Ngo, M. N. Dao, S. Durrani, D. H. N. Nguyen, and R. H. Middleton, "Energy efficiency maximization for downlink cloud radio access networks with data sharing and data compression," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 4955–4970, Aug 2018.
- [41] F. Zhuang and V. K. N. Lau, "Backhaul limited asymmetric cooperation for MIMO cellular networks via semidefinite relaxation," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 684–693, Feb 2014.
- [42] P. Luong, F. Gagnon, C. Despins, and L. Tran, "Optimal joint remote radio head selection and beamforming design for limited fronthaul C-RAN," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5605–5620, Nov 2017.

- [43] Q. Shi, M. Razaviyayn, Z. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.
- [44] L.-N. Tran, M. Hanif, A. Tolli, and M. Juntti, "Fast converging algorithm for weighted sum rate maximization in multicell MISO downlink," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 872–875, Dec 2012.
- [45] O. Tervo, L.-N. Tran, and M. Juntti, "Optimal energy-efficient transmit beamforming for multi-user MISO downlink," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5574–5588, Oct. 2015.
- [46] E. Björnson, G. Zheng, M. Bengtsson, and B. Ottersten, "Robust monotonic optimization framework for multicell MISO systems," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2508–2523, May 2012.
- [47] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl., Special Issue on Linear Algebra in Control, Signals and Image Processing*, pp. 193–228, Nov. 1998.
- [48] K.-G. Nguyen, Q.-D. Vu, L.-N. Tran, and M. Juntti, "Energy efficiency fairness for multi-pair wireless-powered relaying systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 357–373, Feb. 2019.
- [49] S. Lacoste-Julien, "Convergence rate of frank-wolfe for non-convex objectives," *ArXiv e-prints*, Jul 2016.
- [50] H. Wai, J. Lafond, A. Scaglione, and E. Moulines, "Decentralized Frank-Wolfe algorithm for convex and nonconvex problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, Nov 2017.
- [51] F. Pedregosa, A. Askari, G. Negiar, and M. Jaggi, "Step-size adaptivity in projection-free optimization," *ArXiv e-prints*, Oct. 2018.
- [52] C. Song, S. Yoon, and V. Pavlovic, "Fast ADMM algorithm for distributed optimization with adaptive penalty," in *Proc. 30th AAAI Conf. Artificial Intelligence (AAAI-16)*, Feb. 2016.
- [53] I. MOSEK ApS, 2014, [Online]. Available: www.mosek.com.
- [54] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proc. the CACSD Conference*, Taipei, Taiwan., 2004. [Online]. Available: <http://users.isy.liu.se/johanl/yalmip>
- [55] A. Wiesel, Y. C. Eldar, and S. Shamai, "Zero-forcing precoding and generalized inverses," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4409–4418, Sept 2008.
- [56] J. Fessler, "Gradient-based methods," <https://web.eecs.umich.edu/~fessler/course/598/I/>, 2019.