Learning to Beamform in Heterogeneous Massive MIMO Networks

Minghe Zhu, Tsung-Hui Chang, *Senior Member, IEEE,* and Mingyi Hong, *Senior Member, IEEE,*

***Abstract*—Finding the optimal beamformers in massive multiple-input multiple-output (MIMO) networks is challenging because of its non-convexity, and conventional optimization based algorithms suffer from high computational costs. Recently, deep learning based methods have been proposed because of their computational efficiency, but they typically can not generalize well when deployed in heterogeneous scenarios where the base stations (BSs) are equipped with different numbers of antennas and have different inter-BS distances. This paper proposes a novel deep learning based beamforming algorithm to address above challenges. Specifically, we consider the weighted sum rate (WSR) maximization problem in multi-input and single-output (MISO) interference channels, and propose a beamforming learning architecture by unfolding a parallel gradient projection algorithm. By leveraging the low-dimensional structures of the optimal beamforming solution, our constructed learning network can be made independent of the numbers of transmit antennas and BSs. Moreover, such a design can be further extended to a cooperative multicell network where users are jointly served by multiple BSs. Numerical results based on both synthetic and ray-tracing channel models show that the proposed neural network can achieve high WSRs with significantly reduced runtime, while exhibiting favorable generalization capability with respect to the antenna number, BS number and the inter-BS distance.**

***Index Terms*—Beamforming, deep neural network, MISO interfering channel, cooperative multicell beamforming.**

I. Introduction

Massive multiple-input multiple-output (MIMO) is an emerging technology that uses arrays with hundreds of antennas simultaneously serving tens of terminals in the same time-frequency resource [2]. Multiuser beamforming based on massive MIMO can provide high spectral efficiency and have been recognized as a key technology for 5G wireless networks [3].

However, there are many challenges in searching for optimal beamforming strategies for effectively improving the performance of wireless communication systems. First and foremost, the computational complexity brought by significantly increased number of base stations (BSs) and transmit antennas is too high to fulfill the latency requirement of 5G applications. Moreover, relying on the massive MIMO and millimeter wave (mmWave) communication technologies [4], a large number of small cells are expected to be deployed with macrocells heterogeneously in ultra-dense cellular networks. There will be different numbers of access points (APs) installed inside the building or BSs located outdoors which are equipped with different numbers of antennas to deal with the complex communication environments. Beamforming optimization for these dense and heterogeneous networks have to jointly consider different network sizes and antenna configurations, making it much harder to search for a proper beamforming solution. In particular, the computational complexity can quickly become prohibitive with the network size, the number of antennas and the number of served users. Therefore, a multiuser beamforming method that is computationally scalable in such heterogeneous networks is highly desired.

A. Related Work

Beamforming optimization has been an active research area in the past two decades [5]. The power minimization based beamforming problems can be well-solved [5] or well-approximated by various convex optimization techniques [6]. On the contrary, the weighted sum rate maximization (WSRM) based beamforming problem is difficult to solve and in fact NP-hard in general [7, 8]. Many suboptimal but computationally efficient beamforming algorithms have been proposed in the literature. For example, the paper [9] proposed the zero-forcing based beamforming based on the generalized matrix inverse theory. Approximation algorithms based on successive convex approximation techniques are proposed in [10, 11] for efficient resource allocation and multiuser beamforming optimization. The inexact cyclic coordinate descent algorithm proposed in [8] relies on the block coordinate descent (BCD) and gradient projection (GP), and can achieve good performance with a low complexity. The celebrated weighted minimum mean square error (WMMSE) algorithm proposed in [12, 13] is based on the equivalence between signal-to-interference-plus-noise ratio (SINR) and MSE, which then is solved by the BCD method. The WMMSE algorithm provides the state-of-the-art performance and therefore is widely benchmarked in the literature. However, all these existing algorithms are iterative in nature, and their complexities quickly increase with the antenna number and network size.

In recent years, machine learning based approaches based on the deep neural network (DNN) have been considered in

Part of the work was presented in IEEE SAM 2020 [1].

T.-H. Chang is the corresponding author.

M. Zhu and T.-H. Chang are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, and the Shenzhen Research Institute of Big Data, Shenzhen 518172, China (e-mail: minghezhu@link.cuhk.edu.cn, tsunghui.chang@ieee.org).

M. Hong is with the Department of Electrical and Computer Engineering, University of Minnesota, Twin cities, MN, USA (e-mail: mhong@umn.edu).

The work of T.-H. Chang is supported by the NSFC, China, under Grant No. 62071409, the Shenzhen Science and Technology Program under Grant No. RCJC20210609104448114, and by Guangdong Provincial Key Laboratory of Big Data Computing.

This work of M. Hong is supported by NSF grant CNS-2003033.

a range of wireless communication applications [14]. The key motivation is to replace the conventional optimization based iterative algorithms with a pretrained neural network (NN) that can accurately approximate the algorithm solution in a fast and low-complexity manner. For instance, in [15], a parallel structured convolutional neural network (CNN) is trained with only geographical location information of transmitters and receivers to learn the optimal scheduling in dense device-to-device wireless networks. In [16, 17], a black-box DNN is trained to approximate the WMMSE solution for optimal power control in the interference channel.

DNN based beamforming schemes have also been proposed for alleviating the computational issues faced in massive MIMO communications. For example, by considering the multiple-input single-output (MISO) broadcast channel, the work [18] proposed a beamforming neural network (BNN) that learns virtual uplink power variables based on the well-known uplink-downlink duality [5] for the power minimization problem and the SINR balancing problem. For the WSRM problem, they proposed a BNN to learn the power variables and Lagrange dual variables based on the optimal beamforming structure. By considering a coordinated beamforming scenario with multiple BSs serving one receiver, the work [19] proposed a black-box DNN to learn the radio-frequency (RF) downlink beamformers directly from the signals received at the distributed BSs during the uplink transmission.

Different from the black-box DNN approach, the deep unfolding (DU) technique [20, 21] can build a learning network based on approximating a known iterative algorithm with finite iterations. Specifically, the learning network has a recurrent neural network (RNN) structure which is composed by recurrent function blocks and imitates the iterative steps of conventional optimization algorithms. For example, the works [22], [23] and [24] respectively unfold the GP algorithm, alternating direction method of multipliers (ADMM) and gradient descent algorithm to build learning networks for MIMO detection. For a single-cell multiuser beamforming problem, the authors in [25] proposed a learning network by unfolding the WMMSE algorithm. To overcome the difficulty of matrix inversion involved in the WMMSE algorithm, they approximate the matrix inversion by its first-order Taylor expansion. Another recent work [26] considered to unfold the WMMSE algorithm to solve the coordinated beamforming problem in MISO interference channels (MISO-ICs). The advantage of these DU methods is that they can leverage the existing algorithm as the guidance to design the learning network. So the parameters to be learned in the learning network can be much less compared to the black-box DNN methods.

Although above machine learning based methods have shown promising performances, their learned networks usually lack good generalization capabilities, that is, they cannot be easily used to optimize a new scenario in which network parameters such as the number of antennas, the number of BSs or the network size are different from the scenarios when the NN is trained. This shortcoming makes the current designs not suitable for heterogeneous wireless environments.

B. Contributions

In this paper, we consider learning-based beamforming designs for WSRM in the MISO-ICs as well as the cooperative multicell networks (see Fig. 1). Our goal is to design computationally efficient beamforming learning networks (BLNs) that can scale and generalize well in heterogeneous networks. Different from the existing works which rely on the WMMSE algorithm, we propose a BLN framework based on the DU technique and by unfolding the simple parallel GP (PGP) algorithm [27], which is referred to as "BLN-PGP" in the paper.

The first key advantage of the proposed BLN-PGP is that it has a parallel structure and the deployed NN is identical for all BSs. This makes BLN-PGP have only few parameters to be trained and can output a high-quality beamforming solution quickly in time. The second advantage is that the complexity of the NN can be made *independent* of both the number of BSs and the number of BS antennas (which is usually large in massive MIMO communications), and thus the dimension of learnable parameters does not increase with the network size and antenna size. Consequently, the proposed BLN-PGP has the third advantage that it has good generalization capability with respect to the cell radius, the number of antennas, and the number of BSs, which means that the proposed BLN-PGP can be easily deployed in heterogeneous networks (where the BSs have different number of antennas and non-uniform inter-BS distances) without the need of re-training. Our specific technical contributions are summarized as follows.

1) We propose a new DU based BLN framework which is composed by recurrent function blocks that mimic the iterative updating steps of the PGP algorithm for solving the WSRM problem. In order to achieve good beamforming performance within a small number of iterations, we employ a multi-layer perception (MLP) to predict the gradient vector with respect to the beamforming vectors. By showing that the gradient vector lies in a low-dimensional subspace, the MLP simply learns the coefficients required to construct the gradient. Since the MLP is identical for all BSs, the parameter space to be learned is small.
2) We make two key steps to improve the generalization capability of BLN-PGP with respect to the number of transmit antennas and BSs. Firstly, the low-dimensional structure of the optimal beamforming solution is exploited to transform the target WSRM problem into a dimension-reduced one. This ensures that the proposed BLN-PGP solves a problem whose dimension is independent of the number of transmit antennas. Secondly, instead of considering the interference from the whole network, the MLP only considers the signals and interference that are sufficiently strong to predict the beamforming gradient vector. This makes the MLP dimension independent of the network size, and the BLN-PGP can be resilient to the number of BSs. Furthermore, we show that the BLN-PGP can be robust against channel state information (CSI) errors by considering a robust training loss function.
3) The designs of the BLN-PGP are further extended to

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

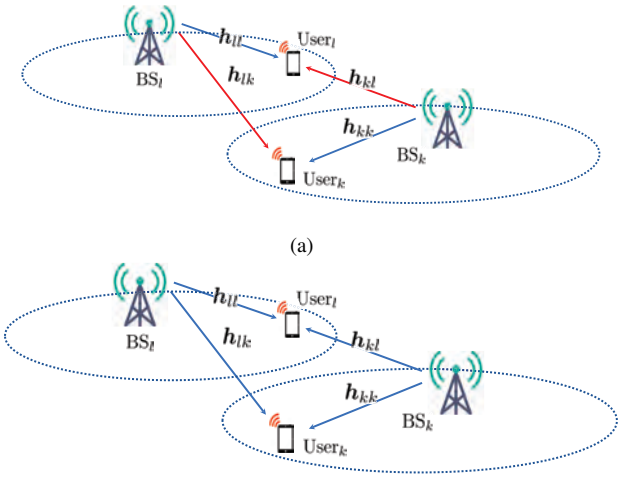


Fig. 1: (a) The MISO-IC [8], and (b) the cooperative multicell network [29]. The blue arrows represent information signals and the red arrows represent the inter-cell interference. Such scenarios occur when the licensed spectrum is used by one operator in a geographic area and the BSs for that operator are coordinated by the cellular core network [30].

the cooperative multicell beamforming problem, which is a joint transmission scheme with users served by all cooperative BSs. The performance of the proposed BLN-PGP is examined by both synthetic channel dataset and ray-tracing based DeepMIMO dataset [28]. The results show that the proposed BLN-PGP can perform comparably with the WMMSE algorithm but has a significantly reduced computation time. More impressively, BLN-PGP shows promising generalization capability with respect to the BS number, antenna number, and inter-BS distance.

Synopsis: Section II presents the system model of the MISO-IC and formulates the WSRM problem. The existing beamforming algorithms are also briefly reviewed. Section III presents the main design of the proposed BLN-PGP, including introduction of the approach to improving its generalization capability. In Section IV, we extend our BLN-PGP to solve the more complex cooperative multicell beamforming problem. The simulation results are given in Section V and the paper is concluded in Section VI.

Notations: Column vectors and matrices are respectively written in boldfaced lower-case and upper-case letters, e.g., $\boldsymbol{a}$ and $\mathbf{A}$, respectively. The superscripts $(\cdot)^{\mathsf{T}}$, $(\cdot)^*$ and $(\cdot)^{\mathsf{H}}$ represent the transpose, conjugate and hermitian transpose respectively. $\boldsymbol{I}_K$ is the $K \times K$ identity matrix; $\|\boldsymbol{a}\|$ denotes the Euclidean norm of vector $\boldsymbol{a}$. $\Im(\cdot)$ and $\Re(\cdot)$ represent the imaginary and real part of a complex value respectively. $\{a_{jk}\}$ denotes the set of all a_{jk} with subscripts j, k covering all the admissible intergers, $\{a_{jk}\}_k$ denotes the set of all a_{jk} with the first subscript equal to j.

II. WSRM Problem and Algorithms

As shown in Fig. 1(a), we first consider the downlink multi-user MISO-IC which has K BSs serving K respective user equipment (UE) at the same time and over the same spectrum. Extension to the cooperative multi-cell scenario shown in Fig. 1(b) will be presented in Section IV. We assume that each BS is equipped with N_t transmit antennas[1], and each UE has only one receive antenna. Let $s_k \in \mathbb{C}$, $\mathbb{E}[|s_k|^2] = 1$, be the information signal for UE$_k$, and $\boldsymbol{v}_k \in \mathbb{C}^{N_t}$ denote the beamforming (BF) vector used by BS$_k$, for all $k \in \mathcal{K} \coloneqq \{1, \ldots, K\}$. Moreover, denote $\boldsymbol{h}_{jk} \in \mathbb{C}^{N_t}$ as the channel between BS$_j$ and UE$_k$. Then, the signal received by UE$_k$ is given by

$$y_k = \boldsymbol{h}_{kk}^{\mathsf{H}} \boldsymbol{v}_k s_k + \sum_{j=1, j\neq k}^{K} \boldsymbol{h}_{jk}^{\mathsf{H}} \boldsymbol{v}_j s_j + n_k, \ k \in \mathcal{K}, \tag{1}$$

where $n_k \in \mathbb{C}$ is the additive white Gaussian noise (AWGN) with zero mean and variance σ_k^2, i.e., $n_k \sim \mathcal{CN}(0, \sigma_k^2)$. It is assumed that the signals s_k, $k \in \mathcal{K}$, are statistically independent from each other and from the AWGN. Thus, the SINR for each UE$_k$ can be written as

$$\mathsf{SINR}_k(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}_j) = \frac{|\boldsymbol{h}_{kk}^{\mathsf{H}} \boldsymbol{v}_k|^2}{\sum_{j\neq k} |\boldsymbol{h}_{jk}^{\mathsf{H}} \boldsymbol{v}_j|^2 + \sigma_k^2}, \ k \in \mathcal{K}. \tag{2}$$

Assume that each BS$_k$ has perfect knowledge of CSI $\{\boldsymbol{h}_{kj}\}_j$, through, e.g., channel estimation schemes in [31]. Then the downlink transmission rate of link k can be expressed as

$$R_k(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}_j) = \log_2\left(1 + \mathsf{SINR}_k(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}_j)\right). \tag{3}$$

We are interested in designing the beamformers so that the network throughput is maximized. Specifically, the WSRM problem is formulated as

$$\begin{aligned} \max_{\boldsymbol{v}_k \in \mathbb{C}^{N_t}, k\in\mathcal{K}} \quad & R(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}) \\ \text{s.t.} \quad & \|\boldsymbol{v}_k\|^2 \leq P_k, k \in \mathcal{K}, \end{aligned} \tag{4}$$

where

$$R(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}) = \sum_{k=1}^{K} \alpha_k \cdot R_k(\{\boldsymbol{v}_k\}, \{\boldsymbol{h}_{jk}\}_j), \tag{5}$$

in which $\alpha_k \geq 0$ is a non-negative weighting coefficient of link k, and P_k denotes the maximum power budget of BS$_k$. Hence, the WSRM problem in the form of (4) has to be solved before the BSs transmit signals to their receivers. However, (4) is a non-convex problem, and it has been shown to be NP-hard in general [7, 8]. In view of this, suboptimal but computationally efficient algorithms have been proposed for (4). Next, we review the WMMSE algorithm [13], the GP algorithm [8, 27], and the polyblock outer approximation (POA) algorithm [32].

A. WMMSE Algorithm

The WMMSE algorithm [13] is one of the most popular algorithms for handling the WSRM problem in (4). It reformulates (4) as an equivalent weighted MSE minimization problem by the MMSE-SINR equality [33], followed by solving the problem with the BCD method [34]. The iterative steps of WMMSE are given by: for iteration $r = 1, \ldots,$ perform

$$u_k^r = \left(\sum_{j=1}^{K} |\boldsymbol{h}_{jk}^{\mathsf{H}} \boldsymbol{v}_j^{r-1}|^2 + \sigma_k^2\right)^{-1} \boldsymbol{h}_{kk}^{\mathsf{H}} \boldsymbol{v}_k^{r-1}, \tag{6a}$$

[1]Here, we made the assumption for ease of problem formulation. The presented algorithms and BLNs are directly applicable to the scenario where the BSs have different number of antennas; see discussions in Remark 2.

$$w_k^r = \left(1 - u_k^r \boldsymbol{h}_{kk}^{\mathsf{H}} \boldsymbol{v}_k^{r-1}\right)^{-1}, \tag{6b}$$

$$\boldsymbol{v}_k^r = \alpha_k \left(\sum_{j=1}^{K} \alpha_j |u_j^r|^2 w_j^r \boldsymbol{h}_{kj} \boldsymbol{h}_{kj}^{\mathsf{T}} + \mu_k^* \boldsymbol{I}_N\right)^{-1} u_k^r w_k^r \boldsymbol{h}_{kk}, \tag{6c}$$

for all $k \in \mathcal{K}$. In (6c), μ_k^* is an optimal dual variable associated with the power budget constraint [13]. Theoretically, it has been shown that the WMMSE algorithm can converge to a stationary solution of (4), and practically performs well.

B. Gradient ascent based Algorithm

Since the power budget constraints of problem (4) have a simple structure, gradient ascent based methods, such as the gradient projection (GP) method [27], can be applied. For example, the inexact cyclic coordinate descent (ICCD) algorithm proposed in [8] deals with problem (4) by applying gradient projection update for each beamformer $\boldsymbol{v}_k$ in a sequential and cyclic fashion. Specifically, the ICCD algorithm has the following steps: for iteration $r = 1, 2, \ldots$, perform for $k = 1, \ldots, K$ sequentially

$$\begin{aligned} \tilde{\boldsymbol{v}}_k^r &= \boldsymbol{v}_k^{r-1} + s_k^{r-1} \nabla_{\boldsymbol{v}_k} R(\{\boldsymbol{v}_j^r\}_{j<k}, \{\boldsymbol{v}_j^{r-1}\}_{j\geq k}, \{\boldsymbol{h}_{jk}\}), \\ \boldsymbol{v}_k^r &= \frac{\tilde{\boldsymbol{v}}_k^r}{\max\{\|\tilde{\boldsymbol{v}}_k^r\|/\sqrt{P_k}, 1\}}, \end{aligned} \tag{7}$$

where $s_k^r > 0$ is the step size, and the step in (7) is projected onto the power constraint in (4). A key advantage of gradient ascent based methods is that they involve simple computation steps. However, compared with the WMMSE algorithm, the gradient ascent based methods usually require a larger number of iterations to converge to a solution as good as the WMMSE solution.

C. POA Algorithm

The POA algorithm [35] is a monotonic optimization method which can asymptotically solve problem (4) to the global solution. Specifically, in POA algorithm, a sequence of surrogate problems are systematically constructed, whose feasible set contains that of the original problem. The constructed feasible set will shrink iteratively and converge to the true feasible set of the original problem [36], while the objective values of the constructed problems will converge to the true optimal value from above asymptotically. Therefore, POA algorithm provides an upper bound solution for the original optimization problem (4). However, the POA algorithm suffers from quite high computation complexity, making it impractical to be used in real-time scenarios.

Remark 1. It is worth noting that all of the above algorithms are iterative in their original form. Therefore, all of them suffer from significant computational delays especially for massive MIMO scenarios where the numbers of cells and transmit antennas are large. To alleviate the computation issues, researchers have proposed the use of DNN [19] and the deep unfolding techniques [20, 25, 26] for approximating the WMMSE beamforming solution in a computationally efficient fashion. However, both beamforming designs in [25, 26] have limited generalization capability with respect to the number of transmit antennas or network size. In particular, the learning networks have to be retrained whenever they are deployed in a scenario with different numbers of BSs and transmit antennas.

III. Proposed Beamforming Learning Network

Our goal is to design a new BLN which has improved generalization capability over the existing DNN based approaches. In this section, we first utilize the low-dimensional structure of the beamforming solution of problem (4) to transform the problem into an equivalent problem with reduced dimension. Then, by unfolding the PGP method, we develop the proposed BLN with a light computation complexity, and present approaches to enhance its generalization capability with respect to various system parameters.

A. Problem Dimension Reduction

Since in massive MIMO scenarios the number of transmit antennas N_t is large, it is desirable to avoid handling problem (4) in its original form. It has been shown in [37, Proposition 1] that the optimal beamforming vectors actually have a low-dimensional structure, as stated below.

Proposition 1. *[37, Proposition 1] Suppose that $N_t \geq K$, and that $\{\boldsymbol{h}_{kj}\}_j$ are linearly independent and satisfy*

$$\boldsymbol{h}_{kj}^{\mathsf{H}} \boldsymbol{h}_{kj'} \neq 0, \ \forall j, j' \in \mathcal{K}, j \neq j'.$$

Then, if $\boldsymbol{v}_k$ is a beamforming vector that corresponds to a rate point on the Pareto boundary, there exist $\{\xi_{kj}\}_{j=1}^K$ such that

$$\boldsymbol{v}_k = \sum_{j=1}^{K} \xi_{kj} \boldsymbol{h}_{kj}, \ \|\boldsymbol{v}_k\|^2 = P_k. \tag{8}$$

By this property, the BF vector for each BS_k lies in the low-dimensional subspace spanned by the channel vectors $\{\boldsymbol{h}_{kj}\}_j$. Let $\mathbf{H}_k = \left[\boldsymbol{h}_{k1}, \ldots, \boldsymbol{h}_{kK}\right] \in \mathbb{C}^{N_t \times K}$ and $\boldsymbol{\xi}_k = \left[\xi_{k1}, \ldots \xi_{kK}\right]^\top \in \mathbb{C}^K$. We can let $\boldsymbol{v}_k = \mathbf{H}_k \boldsymbol{\xi}_k$ for all $k \in \mathcal{K}$, and rewrite problem (4) as

$$\begin{aligned} \max_{\boldsymbol{\xi}_k \in \mathbb{C}^K, k \in \mathcal{K}} \ & \sum_{k=1}^{K} \alpha_k \log_2 \left(1 + \frac{\left|\boldsymbol{h}_{kk}^{\mathsf{H}} \mathbf{H}_k \boldsymbol{\xi}_k\right|^2}{\sum_{j \neq k} \left|\boldsymbol{h}_{jk}^{\mathsf{H}} \mathbf{H}_j \boldsymbol{\xi}_j\right|^2 + \sigma_k^2}\right) \\ \text{s.t.} \ & \|\mathbf{H}_k \boldsymbol{\xi}_k\|^2 \leq P_k, k \in \mathcal{K}. \end{aligned} \tag{9}$$

To avoid handling the ellipsoid constraint $\|\mathbf{H}_k \boldsymbol{\xi}_k\|^2 \leq P_k$, we consider eigen-decomposition of

$$\mathbf{H}_k^{\mathsf{H}} \mathbf{H}_k = \mathbf{U}_k \boldsymbol{\Lambda}_k \mathbf{U}_k^{\mathsf{H}},$$

where $\mathbf{U}_k \in \mathbb{C}^{K \times K}$ is the unitary eigen-matrix and $\boldsymbol{\Lambda}_k \in \mathbb{R}^{K \times K}$ is a diagonal eigenvalue matrix. By letting $\boldsymbol{w}_k = \boldsymbol{\Lambda}_k^{1/2} \mathbf{U}_k^{\mathsf{H}} \boldsymbol{\xi}_k \in \mathbb{C}^K$ and $\boldsymbol{g}_{jk} = \boldsymbol{\Lambda}_j^{-1/2} \mathbf{U}_j^{\mathsf{H}} \mathbf{H}_j^{\mathsf{H}} \boldsymbol{h}_{jk} \in \mathbb{C}^K$, we write (9) as

$$\begin{aligned} \max_{\boldsymbol{w}_k \in \mathbb{C}^K, k \in \mathcal{K}} \ & \sum_{k=1}^{K} \alpha_k \log_2 \left(1 + \frac{\left|\boldsymbol{g}_{kk}^{\mathsf{H}} \boldsymbol{w}_k\right|^2}{\sum_{j \neq k} \left|\boldsymbol{g}_{jk}^{\mathsf{H}} \boldsymbol{w}_j\right|^2 + \sigma_k^2}\right) \\ \text{s.t.} \ & \|\boldsymbol{w}_k\|^2 \leq P_k. \ k \in \mathcal{K}. \end{aligned} \tag{10}$$

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

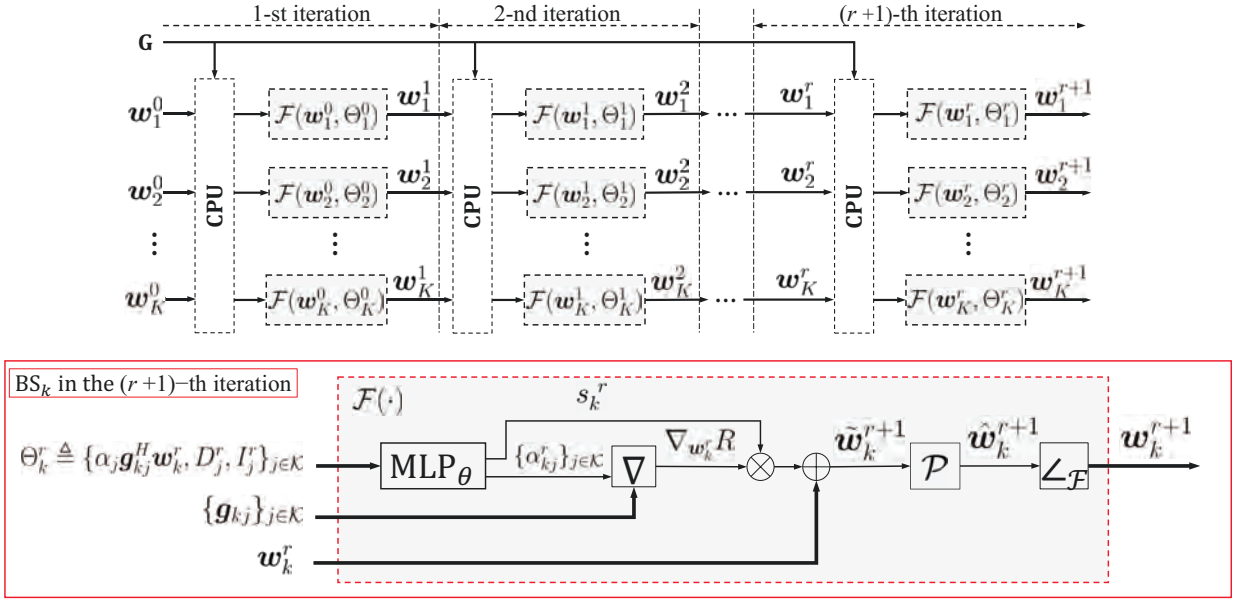


Fig. 2: Diagram of the proposed BLN-PGP for the WSRM problem (10), where $\mathbf{G} = \{\boldsymbol{g}_{jk}\}_{j,k\in\mathcal{K}}$ and $\mathbf{G}_k = \{\boldsymbol{g}_{kj}\}_{j\in\mathcal{K}}$ are CSI matrices. The subscript $(\cdot)_k^r$ refers to the parameter for the k-th BS in the r-th iteration. ∇, $\mathcal{P}$ and $\angle_{\mathcal{F}}$ indicate the gradient, projection and phase rotation respectively.

Comparing (10) with (4), one can see that $\{\boldsymbol{w}_k\}$ are dimension-reduced BF (DRBF) vectors while $\{\boldsymbol{g}_{jk}\}$ are equivalent CSI vectors. In particular, the number of unknown parameters are reduced from $O(KN_t)$ to $O(K^2)$. Therefore, when $N_t \gg K$, it is beneficial to deal with the dimension-reduced problem (10). In addition, problem (10) is independent of N_t, therefore a learning network based on (10) inherently has a good generalization capability with respect to the number of transmit antennas.

B. PGP Inspired Beamforming Learning Network

While the WMMSE algorithm can provide state-of-the-art performance, the matrix inversion structure of the updating rule makes it difficult to build a learning network to learn the beamforming solution [25, 26]. In this section, in view of the separable power constraint, we consider the simple PGP method [27] to handle problem (10). Moreover, based on the DU technique, we show how an effective and computationally efficient BLN can be built.

The PGP method for (10) entails the following steps: for iteration $r = 1, \ldots,$ perform for $k = 1, \ldots, K$ in parallel

$$\tilde{\boldsymbol{w}}_k^r = \boldsymbol{w}_k^{r-1} + s_k^{r-1}\nabla_{\boldsymbol{w}_k} R(\{\boldsymbol{w}_j^{r-1}\}, \mathbf{G}), \tag{11}$$

$$\boldsymbol{w}_k^r = \frac{\tilde{\boldsymbol{w}}_k^r}{\max\{\|\tilde{\boldsymbol{w}}_k^r\|/\sqrt{P_k}, 1\}}, \tag{12}$$

where $\mathbf{G} = \{\boldsymbol{g}_{jk}\}_{j,k\in\mathcal{K}}$ contains all the transformed channel information. According to the DU idea, one can build an NN with a finite iteration number to imitate the iterative updates of the PGP method (11)-(12). In the existing works such as [26], only the step sizes $\{s_k^{r-1}\}$ are set to the learnable parameters. Here, we attempt to learn both the step size s_k^{r-1} and the gradient vector $\nabla_{\boldsymbol{w}_k} R(\{\boldsymbol{w}_j^{r-1}\}, \mathbf{G})$, to find good ascent directions that expedite the algorithm convergence.

It is interesting to note that the gradient vector $\nabla_{\boldsymbol{w}_k} R(\{\boldsymbol{w}_j^{r-1}\}, \mathbf{G})$ in fact lies in the range space of the equivalent CSI vector $\{\boldsymbol{g}_{kj}\}_j$. Specifically, one can have

$$\nabla_{\boldsymbol{w}_k} R(\{\boldsymbol{w}_j\}, \mathbf{G}) = \sum_{j=1}^{K} a_{kj}\mathbf{G}_{kj}, \tag{13}$$

where

$$a_{kk} = \alpha_k \left(\sum_{l=1}^{K} |\boldsymbol{g}_{lk}^{\mathsf{H}}\boldsymbol{w}_l|^2 + \sigma_k^2\right)^{-1} \boldsymbol{g}_{kk}^{\mathsf{H}}\boldsymbol{w}_k, j = k, \tag{14a}$$

$$a_{kj} = \frac{-\alpha_j |\boldsymbol{g}_{jj}^{\mathsf{H}}\boldsymbol{w}_j|^2 \cdot (\boldsymbol{g}_{kj}^{\mathsf{H}}\boldsymbol{w}_k)}{\left(\sum_{l=1}^{K} |\boldsymbol{g}_{lj}^{\mathsf{H}}\boldsymbol{w}_l|^2 + \sigma_j^2\right)\left(\sum_{l\neq j} |\boldsymbol{g}_{lj}^{\mathsf{H}}\boldsymbol{w}_l|^2 + \sigma_j^2\right)}, j \neq k. \tag{14b}$$

In view of (13), it is sufficient to build a learning network that simply learns the K coefficients $\{a_{kj}\}_j$ rather than $\nabla_{\boldsymbol{w}_k} R$.

As shown in Fig. 2, we construct a beamforming learning network termed "BLN-PGP" based on the above ideas. The learning network has a recurrent structure, where each iteration mimics the PGP update (11)-(12) for solving problem (10). Specifically, in each iteration r, it contains K identical function blocks $\mathcal{F}$ that produce the DRBF solutions $\{\boldsymbol{w}_k^r\}_{k\in\mathcal{K}}$ in parallel. Here, we illustrate the detailed operations of the learning network.

In each iteration $r+1$, a central processing unit (CPU) first gathers the channel information $\{\boldsymbol{g}_{kj}\}$, the noise variances $\{\sigma_k^2\}$ and the DRBF vectors $\{\boldsymbol{w}_k^{r-1}\}$ obtained in the previous iteration, and then calculates for each BS$_k$

$$I_j^r = \sum_{l\neq j} |\boldsymbol{g}_{lj}^{\mathsf{H}}\boldsymbol{w}_l^r|^2 + \sigma_j^2, \ j \in \mathcal{K}, \tag{15}$$

$$D_j^r = \alpha_j |\boldsymbol{g}_{jj}^{\mathsf{H}}\boldsymbol{w}_j^r|^2, \ j \in \mathcal{K}, \tag{16}$$

and $\{\alpha_j \boldsymbol{g}_{kj}^{\mathsf{H}}\boldsymbol{w}_k^r\}_{j\in\mathcal{K}}$. The terms D_j^r and I_j^r stand for the information signal power and the suffered interference plus noise power of each UE$_j$, respectively; while $\boldsymbol{g}_{kj}^{\mathsf{H}}\boldsymbol{w}_k^r$ is the

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

out-going interference of BS_k to UE_j. The computed $\Theta_k^r \triangleq \{D_j^r, I_j^r, \alpha_j \boldsymbol{g}_{kj}^{\mathsf{H}} \boldsymbol{w}_k^r\}_{j\in\mathcal{K}}$ together with $\boldsymbol{w}_k^r$ and $\{\boldsymbol{g}_{kj}\}_{j\in\mathcal{K}}$ are used as the input of the main function block $\mathcal{F}$ which outputs $\boldsymbol{w}_k^{r+1} = \mathcal{F}(\boldsymbol{w}_k^r, \Theta_k^r, \mathbf{G}_k)$ as the DRBF vector for BS_k. Such K function blocks are run in a fully parallel fashion. As shown at the bottom of Fig. 2, the function block $\mathcal{F}$ contains the following four key steps.

Gradient prediction: The computed Θ_k^r is used as the input of a multilayer perception (MLP) block associated with each BS_k, which is trained to predict the complex coefficients $\{a_{kj}^r\}_j$ in (13) and the step size s_k^r. The predicted $\{a_{kj}^r\}_j$ are used to construct the gradient vector through

$$\nabla_{\boldsymbol{w}_k^r} R = \nabla\left(\{a_{kj}^r\}_k, \{\boldsymbol{g}_{kj}\}_j\right) = \sum_{j=1}^{K} a_{kj}^r \boldsymbol{g}_{kj}. \tag{17}$$

Note that the MLP in $\mathcal{F}$ are identical for all K BSs, which has common parameters $\boldsymbol{\theta} \in \mathbb{R}^M$, where M is the model size.

Gradient ascent: With $\nabla_{\boldsymbol{w}_k} R^r$ and s_k^r, the gradient ascent update is performed as

$$\tilde{\boldsymbol{w}}_k^{r+1} = \boldsymbol{w}_k^r + s_k^r \nabla_{\boldsymbol{w}_k} R^r.$$

This step is shown in the middle of the enlarged rectangle in Fig. 2, where $\otimes$ and $\oplus$ represent the multiplication and addition operations.

Projection: Following (12), the function block $\mathcal{P}$ projects the coefficient vector $\tilde{\boldsymbol{w}}_k^{r+1}$ onto the feasible set by computing

$$\hat{\boldsymbol{w}}_k^{r+1} = \mathcal{P}(\tilde{\boldsymbol{w}}_k^{r+1}) = \frac{\tilde{\boldsymbol{w}}_k^{r+1}}{\max\{\|\tilde{\boldsymbol{w}}_k^{r+1}\|/\sqrt{P_k}.1\}}. \tag{18}$$

Phase rotation: It is worth noting that problem (10) does not have a unique solution. In fact, if $\{\boldsymbol{w}_k\}$ is an optimal solution of (10), then any phase rotated solution $\{\boldsymbol{w}_k e^{i\psi_k}\}$ is also an optimal solution, where $i = \sqrt{-1}$. In order to make sure that the BLN learns a one-to-one mapping, we rotate the phases of $\{\hat{\boldsymbol{w}}_k^{r+1}\}$ so that each rotated $\hat{\boldsymbol{w}}_k^{r+1}$, denoted by $\boldsymbol{w}_k^{r+1}$, aligns with $\boldsymbol{g}_{kk}$ (i.e., $\boldsymbol{g}_{kk}^{\mathsf{H}} \boldsymbol{w}_k^{r+1}$ is real). In particular, we perform via the function block $\angle_{\mathcal{F}}$

$$\begin{aligned} \boldsymbol{w}_k^{r+1} &= \angle_{\mathcal{F}}(\hat{\boldsymbol{w}}_k^{r+1}) \\ &= \hat{\boldsymbol{w}}_k^{r+1} \exp\left(-i \tan^{-1}\left(\frac{\Im(\boldsymbol{g}_{kk}^{\mathsf{H}} \hat{\boldsymbol{w}}_k^{r+1})}{\Re(\boldsymbol{g}_{kk}^{\mathsf{H}} \hat{\boldsymbol{w}}_k^{r+1})}\right)\right). \end{aligned} \tag{19}$$

Since both $\exp(\cdot)$ and $\tan(\cdot)$ are differentiable functions, the MLP parameter $\boldsymbol{\theta}$ can be trained via the standard backpropagation method; see Section III-D.

As shown in Fig. 2, after a total of T iterations, the BLN outputs the DRBF vectors $\{\boldsymbol{w}_k^T\}$. Then, one can recover the BF vectors $\{\boldsymbol{v}_k^T\}$ of the original problem (4) simply by the transformation $\boldsymbol{v}_k^T = \mathbf{H}_k(\boldsymbol{\Lambda}_k^{1/2}\mathbf{U}_k^{\mathsf{H}})^{-1}\boldsymbol{w}_k^T,\ \forall k \in \mathcal{K}$.

C. Enabling Generalization with the Number of BSs

As seen in the previous subsection, the proposed BLN-PGP network would have good generalization capability with respect to the number of transmit antennas N_t since both the MLPs and operations in each iteration do not depend on it. To enable the MLP to have a generalization capability with respect to the different number of BSs, we need the MLP to have its input $\Theta_k^r = \{D_j^r, I_j^r, \alpha_j \boldsymbol{g}_{kj}^{\mathsf{H}} \boldsymbol{w}_k^r\}_{j\in\mathcal{K}}$ and output $\{a_{kj}^r\}_{j\in\mathcal{K}}$ to be independent of the BS number K.

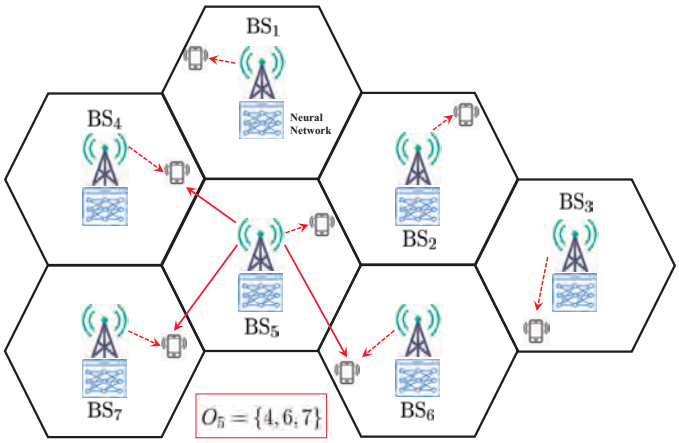


Fig. 3: Illustration of the neighboring "interfered UEs" of BS_5. The solid red arrows represent the interference BS_5 gives to its neighboring "interfered UEs".

As inspired by [38], for each BS_k we define one neighbor subset of UEs whom BS_k causes strong interference to, i.e.,

$$\mathcal{O}_k^r := \left\{j \in \mathcal{K}, j \neq k \middle|\ |\boldsymbol{g}_{kj}^{\mathsf{H}} \boldsymbol{w}_k^r|^2 > \eta \sigma_j^2\right\}, \tag{20}$$

where η a preset threshold. At the beginning of iteration $r+1$, we order $|\boldsymbol{g}_{kj}^{\mathsf{H}} \boldsymbol{w}_k^r|^2/\sigma_j^2,\ j \in \mathcal{O}_k^r$ in a decreasing fashion, and further select the first c indices in $\mathcal{O}_k^r$. The selected subset is denoted by $\mathcal{O}_k^r(c) \subseteq \mathcal{O}_k^r$, which indicates the first c most "interfered" UEs by BS_k. An example is shown in Fig. 3, where $c = 3$, and the "interfered" UEs of it are $\{\text{UE}_4, \text{UE}_6, \text{UE}_7\}$.

Then, we consider only UEs in $\mathcal{O}_k^r(c)$ for computing the input and output of the MLP associated with BS_k. Specifically, instead of Θ_k^r, we let the input of the MLP associated with BS_k be $\Theta_k^r(c) \triangleq \{D_j^r, I_j^r, \alpha_j \boldsymbol{g}_{kj}^{\mathsf{H}} \boldsymbol{w}_k^r\}_{j\in\{\mathcal{O}_k^r(c),k\}}$, In addition to the step size s_k^r, the output of the MLP associated with BS_k is changed to $\{a_{kj}^r\}_{j\in\{\mathcal{O}_k^r(c),k\}}$, which now is dependent on the set $\mathcal{O}_k^r(c)$ only. Therefore, the input and output of the MLP are independent of the whole network size K. The diagram of the revised BLN-PGP network based on $\mathcal{O}_k^r(c)$ is illustrated in Fig. 4, where only the block associated with BS_k at iteration r is plotted.

Remark 2. Given the designs above, the proposed BLN-PGP in Fig. 2 and Fig. 4 have good generalization capability w.r.t. the BS antenna numbers N_t and the cell size K. In fact, since the input and output of the BLN-PGP is independent of the BS antenna numbers, it can be deployed in heterogeneous scenarios where the BSs have different antenna numbers. This also implies that the proposed BLN-PGP needs not to be re-trained when being deployed to a scenario that has antenna numbers and network size different from those when it was trained. This is a significant advantage of BLN-PGP over the existing schemes and will be verified in Section V.

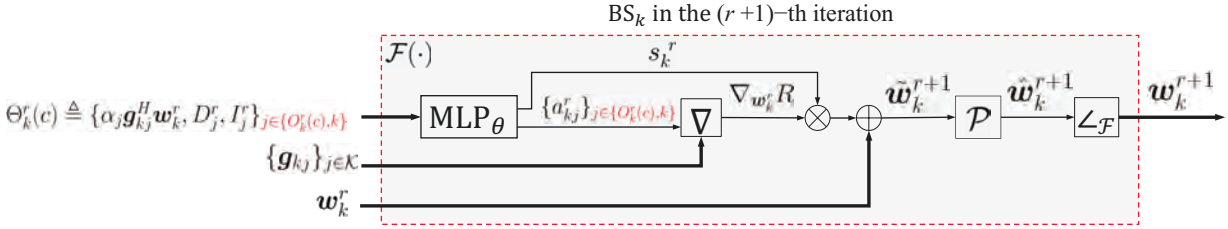


Fig. 4: Diagram of the revised BLN-PGP network for WSRM problem (10) where the MLP input and output are based on the subsets $\mathcal{O}_k^r(c)$. Only the block for BS$_k$ in the $r+1$-th iteration is shown in the figure. ∇, $\mathcal{P}$ and $\angle_{\mathcal{F}}$ indicate the gradient prediction, projection and phase rotation respectively.

D. Hybrid Training Strategy

Suppose that a training data set of size L is given, which contains the CSI $\{\boldsymbol{h}_{jk}^{(\ell)}\}$, WSR coefficients $\{\alpha_k^{(\ell)}\}_{k\in\mathcal{K}}$, and the beamforming solutions $\{\boldsymbol{w}_k^{(\ell)}\}_{k\in\mathcal{K}}$ obtained by an existing algorithm, for $\ell = 1,\ldots,L$. Suppose that the BLN-PGP has a total number of T iterations in the training stage (see Fig. 2 with $r = T$), and denote $\{\boldsymbol{w}_k^{(\ell),r}\}_{k\in\mathcal{K}}$ as the output of BLN-PGP in the rth iteration for the ℓth data sample. The BLN-PGP network is trained by a two-stage approach. The first stage is based on *supervised learning*, using the following loss function

$$\mathrm{L}^{\mathrm{S}}(\boldsymbol{\theta}) = \frac{1}{2LK}\sum_{\ell=1}^{L}\sum_{k=1}^{K}\alpha_k^{(\ell)}\Bigg(\gamma\|\boldsymbol{w}_k^{(\ell)} - \boldsymbol{w}_k^{(\ell),T}\|^2 + (1-\gamma)\sum_{r=1}^{T-1}\|\boldsymbol{w}_k^{(\ell)} - \boldsymbol{w}_k^{(\ell),r}\|^2\Bigg), \tag{21}$$

where γ is the penalty parameter. The second term in the right hand side of (21) encourages the output of earlier iterations of BLN-PGP to be close to $\{\boldsymbol{w}_k^{(\ell)}\}$, which may help speed up the convergence of BLN-PGP.

By treating the supervised training in the first stage as a pre-training, we further refine the network in an *unsupervised fashion* in the second stage. Specifically, we directly train the network so that the WSR function is maximized; for example, we consider the following loss function

$$\mathrm{L}^{\mathrm{U}}(\boldsymbol{\theta}) = -\frac{1}{KL}\sum_{\ell=1}^{L} R(\{\boldsymbol{w}_k^{(\ell),T}\},\{\boldsymbol{h}_{jk}^{(\ell)}\}). \tag{22}$$

As will be shown in Section V, the two-stage training approach can outperform those that solely use supervised training or unsupervised training [18].

E. Improving Robustness under CSI Errors

It is well known that CSI errors at the BSs can degrade the transmission performance, and thereby transmission schemes that are robust to CSI errors are desired in practical systems [39]. In order to improve the robustness of BLN-PGP against the CSI errors, we adopt a simple approach as described below. For each CSI sample $\boldsymbol{h}_{jk}^{\ell}$, we randomly generate E CSI errors $\boldsymbol{e}_{jk}^{(\ell,i)}, i = 1,\ldots,E$, assuming that the CSI errors are bounded and satisfy $\|\boldsymbol{e}_{jk}^{(\ell,i)}\|^2 \le \varepsilon_{jk}\|\boldsymbol{h}_{jk}^{(\ell)}\|^2$, where ε_{jk} is the relative CSI error power. Then, we replace (22) by

$$\mathrm{L}^{\mathrm{U}}(\boldsymbol{\theta}) = -\frac{1}{KLE}\sum_{\ell=1}^{L}\sum_{i=1}^{E} R(\{\boldsymbol{w}_k^{(\ell),T}\},\{\boldsymbol{h}_{jk}^{(\ell)} + \boldsymbol{e}_{jk}^{(\ell,i)}\}). \tag{23}$$

That is, in (23), we consider the average WSR in the presence of the CSI errors, and the BLN-PGP will be trained to improve the average WSR instead.

F. Complexity Comparison

In this subsection, the computational complexity in terms of the number of floating-point operations (FLOPs) for BLN-PGP is analyzed and compared with the PGP algorithm and the WMMSE algorithm. By (11), the number of FLOPs per iteration of the PGP algorithm is $\mathcal{O}(K^2N_t + KN_t + K)$, where $\mathcal{O}(K^2N_t)$ is due to the gradient calculation and $\mathcal{O}(KN_t + K)$ comes from the projection and descent operation. BLN-PGP has a similar complexity as PGP except that the coefficients $\{a_{kj}\}$ in (13) is obtained via the MLP but not by (14). Suppose that the MLP has five layers with the three hidden layers having M_1, M_2 and M_3 neurons, respectively. The complexity of the MLP is given by $\mathcal{O}(cM_1 + M_1M_2 + M_2M_3 + cM_3)$. Thus the complexity of the BLN-PGP per iteration can be computed as $\mathcal{O}(Kc^2 + Kc + cM_1 + M_1N_2 + M_2N_3 + cM_3 + K + c)$, where $Kc^2 + Kc$ is for computing $\Theta_k^r(c)$ and $(K + c)$ is for computing the gradient from the MLP output and gradient projection step. By contrast, the WMMSE has a much higher per-iteration complexity $\mathcal{O}(K^2N_t^2 + K^2N_t + KN_t^{2.37} + K)$ [25], mainly on the matrix inversion operation and updating the dual variables. Since BLN-PGP can converge quickly with only a small number of iterations, the overall computation time of BLN-PGP can be much faster than the PGP algorithm and the WMMSE, which will be demonstrated in Section V.

IV. Extension to Cooperative Multicell Beamforming Problem

The MISO-IC considered in Section II and Section III treats the interference from adjacent cells as noise, resulting in a fundamental limitation on the performance especially for terminals close to cell edges [40]. In recent years, BS coordination has been analyzed as a means of handling inter-cell interference, in which one UE is served by multiple BSs [41].

In this section, we extend the BLN-PGP to the cooperative multicell scenario. As shown in Fig. 1(b), the cooperative multicell communication scenario considered herein consists of K_u single-antenna receivers served by K_t BSs equipped with N_t antennas each. The jth transmitter and kth receiver are denoted BS$_j$ and UE$_k$, respectively; the channel between them is denoted by $\boldsymbol{h}_{jk}$ for $j \in \mathcal{K}_t := \{1,\ldots,K_t\}$ and

$k \in \mathcal{K}_u \coloneqq \{1, \ldots, K_u\}$. Let $\boldsymbol{v}_{jk}$ be the beamforming vector used by BS_j for serving UE_k. The SINR at UE_k is given by

$$\mathsf{SINR}_k = \frac{\left|\sum_{j=1}^{K_t} \boldsymbol{h}_{jk}^{\mathsf{H}} \boldsymbol{v}_{jk}\right|^2}{\sum_{l\neq k}^{K_u} \left|\sum_{j=1}^{K_t} \boldsymbol{h}_{jk}^{\mathsf{H}} \boldsymbol{v}_{jl}\right|^2 + \sigma_k^2}. \tag{24}$$

The WSRM problem is formulated as

$$\max_{\{\boldsymbol{v}_{jk}\}_{j\in\mathcal{K}_t, k\in\mathcal{K}_u}} \quad \sum_{k=1}^{K_u} \alpha_k \log_2\left(1+\mathsf{SINR}_k\right) \tag{25a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K_u} \|\boldsymbol{v}_{jk}\|^2 \le P_j, j \in \mathcal{K}_t, \tag{25b}$$

where (25b) represents the total power constraint of each BS_j.

Analogous to Section III, we employ [42, Theorem 2] to transform problem (25) into a dimension-reduced problem.

Theorem 1. *[42, Theorem 2] For each rate tuple on the Pareto boundary for problem* (25)*, it holds that beamformers* $\{\boldsymbol{v}_{jk}\}$ *that achieve the Pareto boundary fulfill*

$$\boldsymbol{v}_{jk} \in \textit{span}\left(\{\boldsymbol{h}_{jk}\} \bigcup_{l\neq k} \left\{\Pi^{\perp}_{\boldsymbol{h}_{jl}} \boldsymbol{h}_{jk}\right\}\right), \forall j, k, \tag{26}$$

where $\Pi^{\perp}_{\boldsymbol{h}_{jl}} \coloneqq \boldsymbol{I}_{N_t} - \boldsymbol{h}_{jl}\boldsymbol{h}_{jl}^{\mathsf{H}}/\|\boldsymbol{h}_{jl}\|^2$ *is the projection onto the orthogonal complement of* $\boldsymbol{h}_{jl}$.

Based on Theorem 1, the optimal beamforming solution $\boldsymbol{v}_{jk}, j \in \mathcal{K}_t, k \in \mathcal{K}_u$ of problem (25) can be expressed as

$$\boldsymbol{v}_{jk} = \xi_{jk}^k \boldsymbol{h}_{jk} + \sum_{l\neq k}^{K_u} \xi_{jk}^l \boldsymbol{h}_{jk}^{l\perp} = \mathbf{H}_{jk}\boldsymbol{\xi}_{jk}, \tag{27}$$

where $\boldsymbol{h}_{jk}^{l\perp} \coloneqq \Pi^{\perp}_{\boldsymbol{h}_{jl}} \boldsymbol{h}_{jk}$, $\boldsymbol{\xi}_{jk} = \left[\xi_{jk}^1, \ldots, \xi_{jk}^{K_u}\right] \in \mathbb{C}^{K_u}$ and $\mathbf{H}_{jk} = \left[\boldsymbol{h}_{jk}^{1\perp}, \ldots, \boldsymbol{h}_{jk}^{k-1\perp}, \boldsymbol{h}_{jk}, \boldsymbol{h}_{jk}^{k+1\perp}, \ldots, \boldsymbol{h}_{jk}^{K_u\perp}\right] \in \mathbb{C}^{N_t\times K_u}$. Further consider the eigenvalue decomposition of

$$\mathbf{H}_{jk}^{\mathsf{H}}\mathbf{H}_{jk} = \mathbf{U}_{jk}\mathbf{\Lambda}_{jk}\mathbf{U}_{jk}^{\mathsf{H}},$$

and define $\boldsymbol{w}_{jk} = \mathbf{\Lambda}_{jk}^{1/2}\mathbf{U}_{jk}^{\mathsf{H}}\boldsymbol{\xi}_{jk}$ and $\boldsymbol{g}_{jk} = \mathbf{\Lambda}_j^{-1/2}\mathbf{U}_j^{\mathsf{H}}\mathbf{H}_{jk}^{\mathsf{H}}\boldsymbol{h}_{jk}$ as the DRBF vectors and equivalent CSI vectors, respectively. We can rewrite (25) as

$$\max_{\{\boldsymbol{w}_{jk}\}_{j\in\mathcal{K}_t, k\in\mathcal{K}_u}} \sum_{k=1}^{K_u} \alpha_k \log_2\left(1+\frac{\left|\sum_{j=1}^{K_t} \boldsymbol{g}_{jk}^{\mathsf{H}}\boldsymbol{w}_{jk}\right|^2}{\sum_{l\neq k}^{K_u}\left|\sum_{j=1}^{K_t} \boldsymbol{g}_{jk}^{\mathsf{H}}\boldsymbol{w}_{jl}\right|^2 + \sigma_k^2}\right)$$

$$\text{s.t.} \quad \sum_{k=1}^{K_u} \|\boldsymbol{w}_{jk}\|^2 \le P_j, j \in \mathcal{K}_t. \tag{28}$$

Let us slightly abuse the notation by defining $R(\{\boldsymbol{w}_{jk}\}, \mathbf{G})$ as the WSR in (28). The gradient of $R(\{\boldsymbol{w}_{jk}\}, \mathbf{G})$ with respect to each $\boldsymbol{w}_{jk}$ has the following form

$$\nabla_{\boldsymbol{w}_{jk}} R = \sum_{p=1}^{K_u} \alpha_p \left(a_{jp}^{(k)} \boldsymbol{g}_{jp} + b_{jp}^{(k)} \boldsymbol{g}_{jp}^*\right), \tag{29}$$

where $a_{jp}^{(k)} = c_{jp}^{(k)} \boldsymbol{g}_{jp}^{\mathsf{H}}\boldsymbol{w}_{jk}$ and $b_{jp}^{(k)} = c_{jp}^{(k)} \sum_{q\neq j}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{T}}\boldsymbol{w}_{qk}^*$, in which $c_{jk}^{(k)} = \sum_{l=1}^{K_u}\left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qk}^{\mathsf{H}}\boldsymbol{w}_{ql}\right|^2 + \sigma_k^2$ and

$$c_{jp}^{(k)} = \frac{-\left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{qp}\right|^2}{\left(\sum_{l\neq p}^{K_u}\left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{ql}\right|^2 + \sigma_p^2\right)\left(\sum_{l=1}^{K_u}\left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{ql}\right|^2 + \sigma_p^2\right)},$$

for all $p \neq k$. Therefore, similar to Fig. 2, we can build a learning network for problem (28) by unfolding the PGP method and learning the coefficients $\{a_{jp}^{(k)}, b_{jp}^{(k)}\}$ in (29). In Fig. 5, we present the block diagram of the BLN-PGP network for problem (28), where we only plot the function block associated with BS_j at the rth iteration.

The BLN-PGP network for the cooperative multicell scenario has a similar structure as that in Fig. 2. All the MLPs also share the same structure and parameters. The input of the MLP of BS_j for UE_k is $\Theta_{jk}^r \triangleq \{\alpha_p|\boldsymbol{g}_{jp}^{\mathsf{H}}\boldsymbol{w}_{jk}^r|^2, \alpha_p|\sum_{q\neq j}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{qk}^r|^2, D_p^r, I_p^r\}_{p\in\mathcal{K}_u}$, where

$$D_p^r = \left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{qp}^r\right|^2, I_p^r = \sum_{l\neq p}^{K_u}\left|\sum_{q=1}^{K_t} \boldsymbol{g}_{qp}^{\mathsf{H}}\boldsymbol{w}_{ql}^r\right|^2 + \sigma_p^2. \tag{30}$$

The output of the MLP of BS_j for UE_k in the rth iteration is $\{a_{jp}^{(k),r}, b_{jp}^{(k),r}\}_{p\in\mathcal{K}_u}$ and the step size $s_{jk}^{(k),r}$. Then, the block ∇ constructs the gradient vector according to (29), i.e.,

$$\begin{aligned}\nabla_{\boldsymbol{w}_{jk}^r} R &= \nabla(\{a_{jp}^{(k),r}, b_{jp}^{(k),r}\}_p, \{\boldsymbol{g}_{jp}\}_p) \\ &= \sum_{p=1}^{K_u}\left(a_{jp}^{(k),r}\boldsymbol{g}_{jp} + b_{jp}^{(k),r}\boldsymbol{g}_{jp}^*\right), \ k \in \mathcal{K}_u,\end{aligned} \tag{31}$$

followed by gradient ascent update $\tilde{\boldsymbol{w}}_{jk}^{r+1} = \boldsymbol{w}_{jk}^r + s_{jk}^r \nabla_{\boldsymbol{w}_{jk}^r} R$, $k \in \mathcal{K}_u$.

All the DRBF vectors $\{\tilde{\boldsymbol{w}}_{jk}^{r+1}\}_k$ due to BS_j will be collected and used to perform projection onto the feasible set of problem (28), which yields

$$\hat{\boldsymbol{w}}_{jk}^{r+1} = \mathcal{P}(\tilde{\boldsymbol{w}}_{jk}^{r+1}) = \frac{\tilde{\boldsymbol{w}}_{jk}^{r+1}}{\max\{\sqrt{\sum_{k=1}^{K_u}\|\tilde{\boldsymbol{w}}_{jk}^{r+1}\|^2/P_j}, 1\}}, \tag{32}$$

for all $k \in \mathcal{K}_u$. Lastly, the function block $\angle_{\mathcal{F}}$ rotates the phases of $\{\hat{\boldsymbol{w}}_{jk}^{r+1}\}_k$ by

$$\begin{aligned}\boldsymbol{w}_{jk}^{r+1} &= \angle_{\mathcal{F}}(\hat{\boldsymbol{w}}_{jk}^{r+1}) \\ &= \hat{\boldsymbol{w}}_{jk}^{r+1}\exp\left(-i\tan^{-1}\left(\frac{\Im(\boldsymbol{g}_{jk}^{\mathsf{H}}\hat{\boldsymbol{w}}_{jk}^{r+1})}{\Re(\boldsymbol{g}_{jk}^{\mathsf{H}}\hat{\boldsymbol{w}}_{jk}^{r+1})}\right)\right).\end{aligned} \tag{33}$$

To train the BLN-PGP network in Fig. 5, we adopt the same hybrid strategy in Section III-D. Similarly, the BF vectors $\{\boldsymbol{v}_{jk}\}$ of the original problem (25) can be recovered from $\{\boldsymbol{w}_{jk}\}$ by $\boldsymbol{v}_{jk} = \mathbf{H}_{jk}(\mathbf{\Lambda}_{jk}^{1/2}\mathbf{U}_{jk}^{\mathsf{H}})^{-1}\boldsymbol{w}_{jk}$ before being applied to the antennas of the BS.

We remark that, similar to that in Fig. 2, BLN-PGP in Fig. 5 for the cooperative multicell scenario inherently has a good generalization with respect to the number of transmit antennas N_t. Besides, since the input size and output size of the MLPs

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

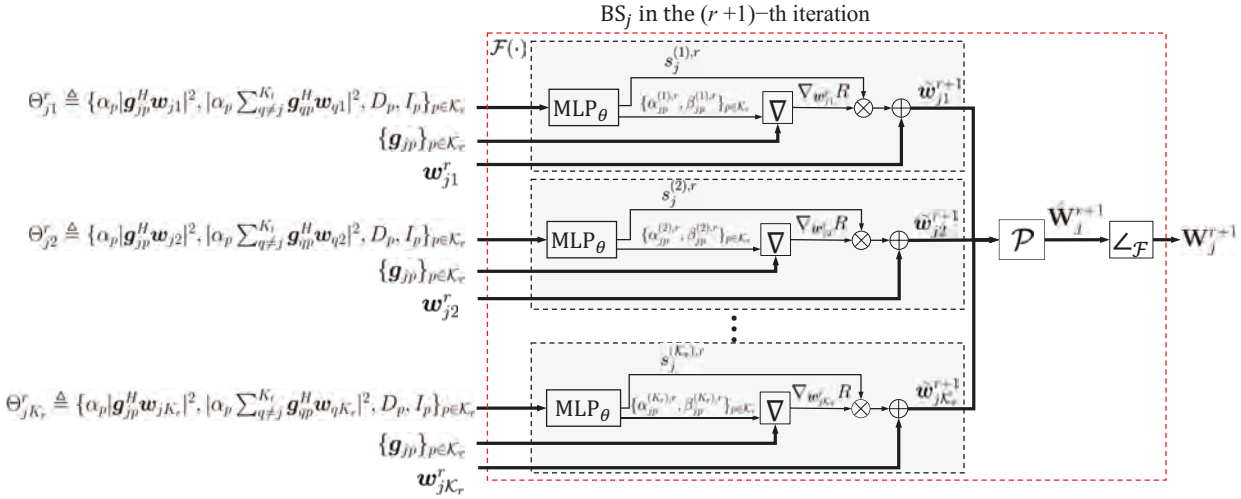


Fig. 5: Diagram of the one block of BS$_j$ of the proposed BLN-PGP network for WSRM problem (28) of the cooperative multicell scenario. The subscript $(\cdot)^{(k),r}_{jp}$ refers to the parameters in the j-th BS for the k-th user in the r-th iteration. $\hat{\mathbf{W}}_j = [\hat{\boldsymbol{w}}_{j1}, \ldots, \hat{\boldsymbol{w}}_{jK_u}]$, $\mathbf{W}_j = [\boldsymbol{w}_{j1}, \ldots, \boldsymbol{w}_{jK_u}]$. The subscript $(\cdot)^r_k$ refers to the parameter for the k-th BS in the r-th iteration. ∇, $\mathcal{P}$ and $\angle_{\mathcal{F}}$ indicate the gradient, projections and phase rotation respectively.

in Fig. 5 do not depend on the number of cooperative BSs K_t, the BLN-PGP network also has good generalization capability with respect to K_t. These will be examined in Section V.

V. Simulation Results

In this section, we present numerical results of the proposed BLN-PGP in Fig. 2 and Fig. 5. Both synthetic channel and the ray-tracing based DeepMIMO dataset [28] are considered .

A. Simulation Setup

We first consider the MISO-IC model in Section II and test the performance of the proposed BLN-PGP in Fig. 2 under the synthetic Rayleigh channel data. Like Fig. 3, we assume that each BS_k is located at the center of cell k and UE_k is located randomly according to a uniform distribution within the cell. The half BS-to-BS distance is denoted as d and set to 1 km if not mentioned specifically. We set P_k, i.e., the maximum transmit power level of BS_k, to be 38 dBm over a 10 MHz frequency band. The carrier frequency is set to 2GHz, and the path loss ρ between the UE and its associated BS is set as $128.1 + 37.6\log_{10}(s)$ (dB) [43], where s (km) is the distance between the UE and BS. The channel coefficients of $\{\boldsymbol{h}_{kj}\}$ are obtained by i.i.d. and standard complex Gaussian random variables scaled by $\sqrt{\rho}$. The noise power spectral density of all UEs are set the same and equal to $\sigma^2 = -174$ dBm/Hz.

Setting in the training stage: A total of 5000 training samples ($L = 5000$) are generated, each of which contains the CSI $\{\boldsymbol{h}_{jk}^{(\ell)}\}$, WSR coefficients $\{\alpha_k^{(\ell)}\}_{k\in\mathcal{K}}$, and the BF solutions $\{\boldsymbol{v}_k^{(\ell)}\}_{k\in\mathcal{K}}$ obtained either by the PGP method or the POA algorithm mentioned in Section II. To train the BLN-PGP network in Fig. 2, we set the parameter η in (20) to be 5. The iteration number of BLN-PGP in the training stage is set to $R = 20$, i.e., $r = 1, \ldots, R$. The function $\tanh$ is used as the activation function in the MLP and the Adam optimizer is used for training the BLN-PGP. The simulation environment is based on TensorFlow 1.14.0 on a desktop computer with Intel i7-9800X CPU Core, one NVIDIA RTX 2080Ti GPU, and 64GB of RAM. The GPU is used during the training stage and CPU is used in the testing stage for all the methods.

Setting in the testing stage: A total of 1000 testing samples are generated in the same way as the training data. If not mentioned specifically, the beamforming solutions of BLN-PGP are obtained by running $T = 20$ iterations, i.e., $r = 1, \ldots, T$, in the testing stage. In the presented experiment results, we evaluate the following methods:

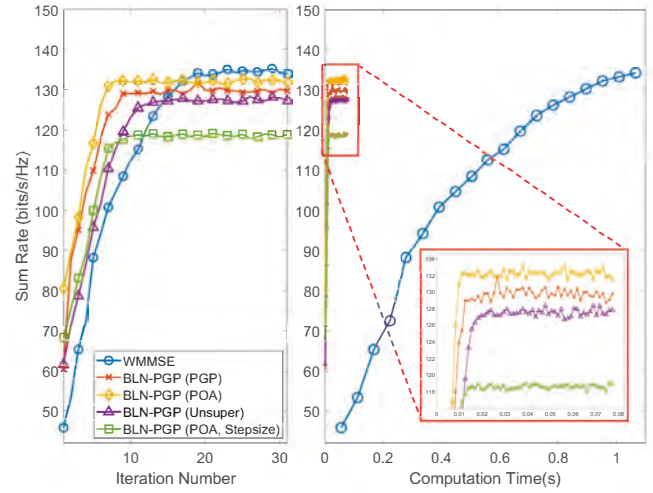
- **PGP, WMMSE and POA:** Performance achieved by the three methods for problem (10).
- **BLN-PGP (PGP) and BLN-PGP (POA):** The BLN-PGP in Fig. 2 trained by the hybrid strategy, and the beamforming solutions used for supervised training are obtained by the PGP and POA methods, respectively.
- **DNN (PGP) and DNN (POA):** The black-box DNNs, which have 5 layers with the concatenated CSI as the input and the beamforming solutions as the output, are trained end-to-end by the hybrid training strategy, and the beamforming solutions used for supervised training are obtained by the PGP and POA methods, respectively.
- **BLN-PGP (Unsuper):** The BLN-PGP in Fig. 2 trained solely by the unsupervised cost.
- **BLN-PGP (POA, Stepsize):** The MLPs in the BLN-PGP only predicts the step size s_k^r, and the gradient vector $\nabla_{\boldsymbol{w}_k} R^r$ is computed explicitly by (13) and (14). The network is trained by the hybrid strategy and the beamforming solutions obtained by the POA method is used during the supervised training.

We also show the "accuracy" (%) which is the ratio of the sum rate achieved by BLN-PGP and that by the WMMSE solution.

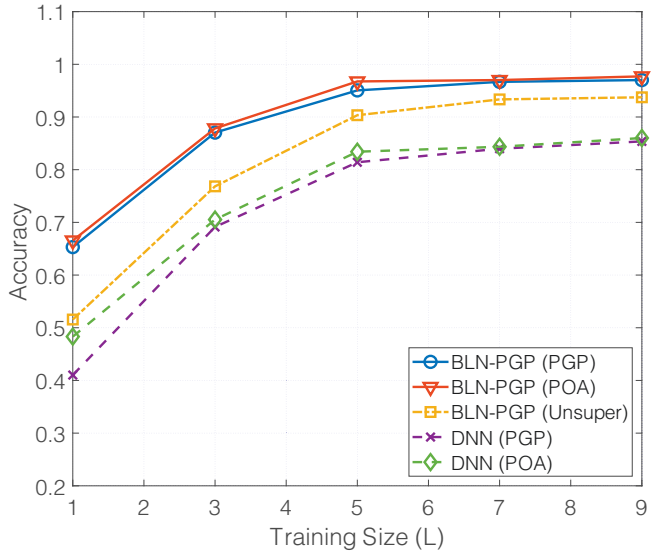
B. Sum-rate Performance

In this subsection, we evaluate the sum-rate performance of different schemes. The results are shown in Fig. 6(a) and Tables I, II and III. For the "BLN-PGP (PGP)", "BLN-PGP (POA)" and "BLN-PGP (Unsuper)", the setting of the MLP is the same as that in Section V-C. For the "DNN (PGP)" and

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662



(a)



(b)

Fig. 6: (a) The sum rates achieved by various schemes versus the iteration number and runtime in the testing stage; The sum rates achieved by "POA" and "DNN (POA)" are 144.57 and 120.59 (bits/s/Hz) respectively. (b) Accuracy achieved by various schemes in the testing stage for different numbers of training samples.

"DNN (POA)", we use a 5-layers DNN where the number of nodes of the input and output layers are both $2KN_t$, and the numbers of nodes in the hidden layers are $1450, 1250, 1325$, respectively. The input of the MLP in the "BLN-PGP (POA, Step-size)" is the same as that of the "BLN-PGP (PGP)", but the numbers of neurons of the hidden layer are $120, 75, 25$, respectively, and the number of neurons of the output is reduced to 1.

Sum-rate versus iteration number and runtime: The experiment results of the achieved sum rates of various schemes versus the iteration number r in the testing stage are shown in the left-side of Fig. 6(a). Firstly, one can observe that the POA algorithm provides an upper bound, and that the "BLN-PGP (POA)" performs slightly poorer than the WMMSE algorithm.

Secondly, we can see that "BLN-PGP (POA)" converges faster and yields higher sum rates than "BLN-PGP (PGP)", "BLN-PGP (Unsuper)" and "BLN-PGP (POA, Stepsize)", which shows the benefits of the hybrid training strategy and prediction of the gradient vector. Note from the figure that both "BLN-PGP (POA)" and "BLN-PGP (PGP)" can converge well around 10 iterations which is much smaller than the WMMSE algorithm. Thirdly, except for "BLN-PGP (POA, Stepsize)", the BLN-PGP can greatly outperform the black-box based "DNN (POA)".

As seen from the right-side of Fig. 6(a), the BLN-PGP schemes have advantage in terms of the runtime. Specifically, for running 20 iterations, the average runtimes of the "BLN-PGP (POA)" "BLN-PGP (PGP)" are about **0.0573s** and **0.0576s**, while the runtimes of the WMMSE and POA algorithms for 20 iterations are 1.964s and 23.231s, respectively.

Impact of training sample size: We examine the achieved accuracy versus the size of training data (L), as is shown in Fig. 6(b). We can see that all schemes can have improved performance when the number of training samples increases. Moreover, we compare the performance of BLN-PGP when different training approaches are used. We can see that there is a gap between hybrid training and unsupervised training, but such a gap reduces when increasing the training data size. This implies that the advantage of hybrid training can be significant if the training size is small. One can also see from the figure that the gap between the black-box based DNN schemes and the proposed BLN-PGP cannot be effectively reduced when the training data size increases.

Generalization w.r.t. number of transmit antennas: To demonstrate the generalization capability of the proposed BLN-PGP, we train the "BLN-PGP (POA)" using the data set of $N_t = 36$ and $K = 19$, but test it on data sets with different numbers of N_t. The results are shown in the 3rd row of Table I. One can see that the proposed BLN-PGP can yield almost the same accuracy when applied to scenarios with $N_t = 36, 72$, and 108. Interestingly, we also test the "BLN-PGP (POA)" in a heterogeneous scenario where the BSs' antenna numbers are randomly chosen from 16 to 128. As seen from the table, "BLN-PGP (POA)" still maintain an average accuracy of 93.78%. In Table I, we also present the results when "BLN-PGP (POA)" is trained by a mixed data set which contains equal-sized data samples with $N_t \in \{18, 36, 72, 108\}$. One can see that this scheme provides slightly higher accuracy.

Generalization w.r.t. number of BSs: To verify the generalization capability of the BLN-PGP with respect to the number of BSs K, we consider a training scenario with $N_t = 64$, $K = 37$ with $c = 18$ and 6, respectively. From Table II(a), one can see that with $c = 18$, the trained "BLN-PGP (POA)" yields a test accuracy of 96.21% for $K = 37$, and has slightly reduced accuracies when deployed in scenarios with $K = 61$ and $K = 91$. We also present in the 4th column of the table the result when the trained BLN-PGP is deployed in a scenario where both K and N_t are respectively changed to 91 and 128. The accuracy is maintained around 92.02%.

In Table II(b), we present another set of results with $c = 6$. One can see that the performance degradation is significant when the trained BLN-PGP is deployed to scenarios with different numbers of BSs. This implies that the neighbor size c considered in the BLN-PGP network training should not be too small when compared to K.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

TABLE I: The sum-rate performance of the proposed BLN-PGP in the testing stage for $K = 19$ and $c = 18$; the size of hidden layers of the MLP are 125, 100, and 85 respectively (MLP 125:100:85).

Number of antennas		$N_t = 36$	$N_t = 72$	$N_t = 108$	randomly $N_t \in [16, 128]$
PGP		134.21	145.16	151.19	-
BLN-PGP (POA)	Trained with $N_t = 36$	129.85 (96.75%)	137.29 (94.58%)	142.81 (94.46%)	- (93.78%)
	Trained with mixed $N_t \in \{18,\ 36,\ 72,\ 108\}$	128.34 (95.63%)	138.47 (95.39%)	143.89 (95.17%)	- (95.27%)

TABLE II: The sum-rate performance of the proposed BLN-PGP in the testing stage for $N_t = 64$.

(a)

Number of Neighbors		$c = 18$ (MLP $85:73:42$)			
Number of BS-user links		$K = 37$	$K = 61$	$K = 91$	$K = 91$ ($N_t = 128$)
BLN-PGP (POA)	Trained with $K = 37$	222.34 (96.21%)	289.92 (93.35%)	537.02 (92.21%)	598.01 (92.02%)
	Trained with $K \in \{37,\ 61,\ 91\}$	220.36 (95.35%)	292.01 (94.02%)	542.83 (93.21%)	601.64 (92.57%)

(b)

Number of Neighbors		$c = 6$ (MLP $32:21:15$)			
Number of BS-user links		$K = 19$	$K = 37$	$K = 61$	$K = 61$ ($N_t = 128$)
BLN-PGP (POA)	Trained with $K = 37$	122.25 (91.09%)	213.79 (92.51%)	275.99 (88.86%)	379.30 (87.69%)
	Trained with $K \in \{19,\ 37,\ 61\}$	124.63 (92.87%)	211.48 (91.51%)	282.70 (91.02%)	391.72 (90.56%)

TABLE III: The sum-rate performance of the proposed BLN-PGP in the testing stage for $K = 19$ and $N_t = 36$.

(a)

Distance Between BSs	$d = 1$ (km)			
Number of Neighbors	$c = 6$ (MLP 32:21:15)	$c = 6$ ($N_t = 128$)	$c = 9$ (MLP 45:38:23)	$c = 18$ (MLP 85:73:42)
Trained with $d = 1$ (km)	128.85 (96.01%)	156.43 (97.29%)	129.63 (96.57%)	129.85 (96.75%)
Trained with $d \in \{0.5,\ 1\}$	128.28 (95.59%)	156.18 (97.11%)	128.65 (95.85%)	128.96 (96.09%)

(b)

Distance Between BSs	$d = 0.5$ (km)			
Number of Neighbors	$c = 6$ (MLP 32:21:15)	$c = 6$ ($N_t = 128$)	$c = 9$ (MLP 45:38:23)	$c = 18$ (MLP 85:73:42)
Trained with $d = 1$ (km)	138.59 (82.73%)	187.65 (97.18%)	146.76 (87.62%)	159.22 (95.05%)
Trained with $d \in \{0.5,\ 1\}$	160.19 (95.63%)	187.78 (97.25%)	160.76 (95.97%)	161.34 (96.32%)

Generalization w.r.t. cell radius: Here, we examine the generalization capability of the BLN-PGP with respect to the cell radius d. We consider a training scenario with $N_t = 36$, $K = 19$ and different number of neighbors $c = 6, 9, 18$, and the half inter-BS distance is fixed to $d = 1$ km or is mixed with $d \in \{0.5, 1\}$.

Table III(a) shows the results that the trained frameworks are tested in the scenario with the same cell radius $d = 1$ km, while Table III(b) are the results obtained when tested in the scenario with $d = 0.5$ km. By comparing the 3rd rows and first columns of the two tables, the accuracy decreases from 96.01% to 82.73% when the trained BLN-PGP is deployed in a scenario with the cell radius decreased to 0.5 km, while the sum rate improvement is minor (from 128.85 to 138.59). This implies that the trained network cannot effectively mitigate the inter-cell interference. Interestingly, as seen from the 2nd columns of the two tables, if we apply the BLN-PGP to scenario with the antenna number increased to $N_t = 128$, the accuracy degradation due to decreased cell radius becomes minor and the sum rate improvement is more evident (from 156.43 to 187.65). By comparing the 4th rows and the first two columns of the two tables, one can see that training with mixed cell radius provides good robustness. Lastly, comparing the first column with the 3rd and 4th columns of Table III(a)-(b), one can also see that larger values of c can make the BLN-PGP to achieve higher accuracy for both $d = 1$ km and $d = 0.5$ km. Therefore, in practice, c should be well selected according to the interference environment of the wireless communication system, such as the number of BS-UE pairs and the distance between BSs and UEs.

C. Impact of the penalty parameter γ

In this subsection, the impact of the penalty parameter γ in (21) on the converge speed of the proposed BLN-PGP algorithm are evaluated. The weights α_k of all users are set to be 1. In the training stage, different values of γ are selected as the penalty parameter in the cost function (21). If not mentioned specifically, we consider the MISO-IC with $K = 19$ and $N_t = 36$. We set the neighbor size c to be 18 ($c = 18$). The MLP used is a 5-layers DNN with the numbers of neurons of the input layer and output layer are $4K$ and $2K + 1$, respectively, and the numbers of neurons in the three hidden layers are $125, 100, 85$, respectively. All the training data and testing data are obtained by the PGP methods, and then the sum-rates in each iteration during the testing stage are demonstrated in Fig. 7.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

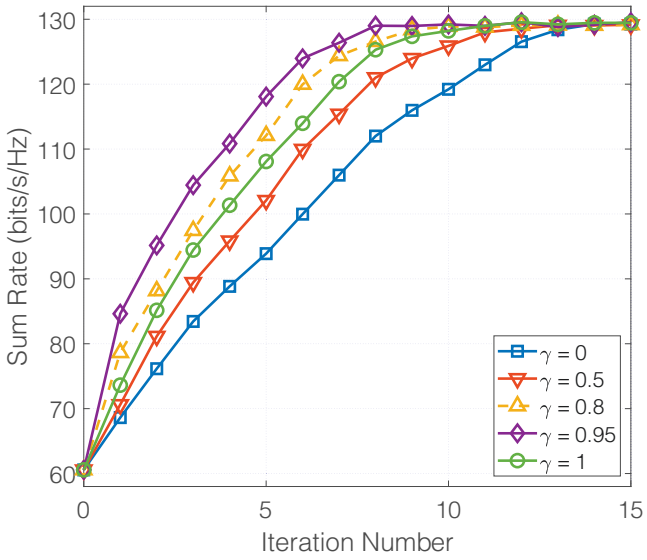


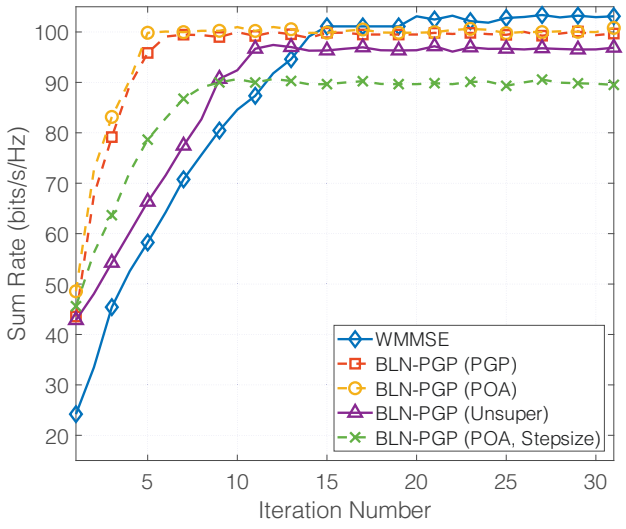
Fig. 7: Sum rates achieved by BLN-PGP (PGP) with various choices of γ in (21) in the testing stage

Fig. 8: The outdoor scenario provided in 'O1' of DeepMIMO dataset [28]

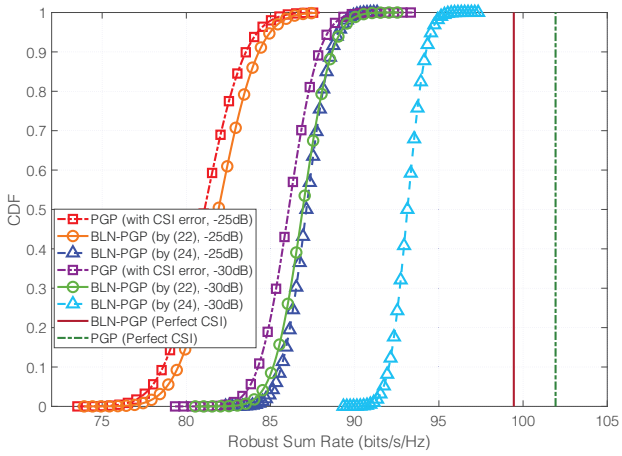
In the Fig. 7, we can see that the choice of γ in the training stage will influence the convergence iteration number in the testing stage. Firstly, we see from Fig. 7 that for all values of γ, the BLN eventually converges to a similar sum rate. Secondly, when compared with that with $\gamma = 1$, the BLN with either $\gamma = 0.95$ or $\gamma = 0.8$ can have improved convergence performance. Thus, we see the acceleration effect. Intriguingly, when $\gamma = 0$, the BLN has the slowest convergence rate. One possible explanation is that the 2nd term of (21) can be regarded as a 'multi-task' function for the BLN so the network is more difficult to train if it is the training objective function. Therefore, it is better for the BLN to primarily focus on improving the quality of the final output (the 1st term of (21)) and take the 2nd term of (21) as a 'regularization'. This intuition is also consistent with what we observe from Fig. 7.

D. Performance on DeepMIMO dataset

In this subsection, we test the proposed BLN-PGP on the ray-tracing based DeepMIMO dataset [28]. We consider an outdoor scenario 'O1' as shown in Fig. 8. The operating carrier



(a)



(b)

Fig. 9: (a) Sum-rate versus iteration number in DeepMIMO channel for $N_t = 36$ $(X = 1, Y = 6, Z = 6)$, $K = 18$ and $c = 17$; The sum rates for "POA", "DNN (PGP)" and "DNN (POA)" are 107.21, 87.72 and 89.41 (bits/s/Hz), respectively. (b) Cumulative distribution of robust sum-rates (MLP $95 : 82 : 48$).

frequency is 3.5 GHz, the bandwidth is 0.5 GHz, number of subcarriers is 64, and the number of multipath is 3. The main street (the horizontal one) is 600m long and 40m wide, and the second street (the vertical one) is 440m long and 40m wide. We consider 18 BSs ($K = 18$), and their served UEs are selected randomly around their respective BS within 50m on the streets (the intersection of the black street and the red circle in Fig. 8). The BSs are equipped with antennas with dimensions X, Y, and Z along the x, y, and z axes, and the antenna size is $N_t = XYZ$. Analogous to the synthetic data, we generated 5000 training samples $(L = 5000)$ and 1000 test samples.

In this experiment, for the proposed BLN-PGP, the numbers of neurons of the 5-layer MLP are $72, 90, 75, 45$ and 37, respectively; the black-box based DNNs also have 5 layers with numbers of neurons equal to $2592, 2845, 2450, 1450$ and 1296, respectively. The testing results versus the iteration number are shown in Fig. 9(a). Again, we can observe consistent results and the proposed BLN-PGP can achieve good performance.

TABLE IV: The sum-rate performance of the proposed BLN-PGP in the testing stage under DeepMIMO dataset [28] for $K = 18$ and $c = 17$ (MLP 90:75:45).

Number of antennas (number of transmit antennas in x, y and z axes)		$N_t = 36$ $(X = 1, Y = 6, Z = 6)$	$N_t = 72$ $(X = 1, Y = 6, Z = 12)$	$N_t = 108$ $(X = 3, Y = 6, Z = 6)$	Random N_t $(X \cdot Y \cdot Z \in [16, 128])$
PGP		102.04	110.36	114.94	-
BLN-PGP	Trained with $X = 1, Y = 6, Z = 6$	98.49 (96.53%)	104.15 (94.37%)	108.18 (94.12%)	- (93.57%)
	Trained with $N_t \in \{36,\ 72,\ 108\}$	97.38 (95.43%)	105.05 (95.19%)	109.22 (95.02%)	- (95.09%)

TABLE V: The sum-rate performance of the proposed BLN-PGP in the testing stage under DeepMIMO dataset [28] for $K_u = K_t$.

(a)

Number of BSs {active BSs}		$K_t = K_u = 6$ (MLP 45:32:28) $\{5, 6, 7, 8, 15, 16\}$		
Number of antennas (number of transmit antennas in x, y and z axes)		$N_t = 36$ $(X = 1, Y = 6, Z = 6)$	$N_t = 64$ $(X = 1, Y = 8, Z = 8)$	$N_t = 216$ $(X = 6, Y = 6, Z = 6)$
BLN-PGP	Trained with $X = 1, Y = 8, Z = 8$	24.56 (92.49%)	28.12 (94.56%)	36.07 (92.47%)
(Trained by PGP)	Trained with $N_t \in \{36,\ 64,\ 216\}$	24.84 (93.52%)	27.80 (93.49%)	36.47 (93.51%)

(b)

Number of BSs {active BSs}		$K_t = K_u = 12$ (MLP 75:56:50) $\{3, 4, 5, 6, 7, 8, 9, 10, 15, 16, 17, 18\}$		
Number of antennas (number of transmit antennas in x, y and z axes)		$N_t = 36$ $(X = 1, Y = 6, Z = 6)$	$N_t = 64$ $(X = 1, Y = 8, Z = 8)$	$N_t = 216$ $(X = 6, Y = 6, Z = 6)$
BLN-PGP	Trained with $X = 1, Y = 8, Z = 8$	37.46 (92.44%)	43.72 (94.23%)	54.99 (92.40%)
(Trained by PGP)	Trained with $N_t \in \{36,\ 64,\ 216\}$	37.88 (93.49%)	43.35 (93.43%)	55.42 (93.13%)

TABLE VI: The sum-rate performance of the proposed BLN-PGP in the testing stage under DeepMIMO dataset [28] for $N_t = 12$, $K_u = 12$ and $N_t = 64$ $(X = 1, Y = 8, Z = 8)$, (MLP 75:56:50).

Number of BSs K_t		$K_t = 6$	$K_t = 12$	$K_t = 18$	Randomly $K_t \in [6, 18]$
PGP		39.47	46.45	69.66	-
BLN-PGP	Trained with $K_t = 12$	36.57 (92.36%)	43.76 (94.23%)	64.21 (92.17%)	- (92.78%)
(Trained by PGP)	Trained with $K_t \in \{6,\ 12,\ 18\}$	36.88 (93.43%)	43.6 (93.89%)	64.96 (93.26%)	- (93.31%)

Similar to Table I, we test the generalization capability of the BLN-PGP in the DeepMIMO data set. As seen from Table IV, the proposed BLN-PGP still can be generalized well and maintain good accuracies.

E. Performance under CSI errors

Herein, we test BLN-PGP under CSI errors. We consider the same setting as Section V-D using the DeepMIMO dataset, except that CSI errors are introduced. Specifically, as described in Section III-E, in the training stage, for each CSI sample $\boldsymbol{h}_{jk}^{\ell}$, 500 random errors $\boldsymbol{e}_{jk}^{(\ell,i)}, i = 1, \ldots, 500$, with $\varepsilon_{jk} \in \{-25, -30\}$ dB, are generated and used in the training loss (23), respectively. In the testing stage, we randomly pick one test sample, and generate 500 CSI error vectors to evaluate the achieved sum rate under the CSI error. In particular, we plot the cumulative density functions (CDFs) in Fig. 9(b). We can see that the CSI errors can cause significant sum rate drops for both PGP and BLN-PGP trained by (21). When the BLN-PGP is trained by (23), we can see from this figure that the sum rate degradation can be reduced about 30% to 40%.

F. Performance for Cooperative Multicell Beamforming

In this subsection, we examine the proposed BLN-PGP in Fig. 5 for the cooperative multicell beamforming problem in Section IV. We again consider the outdoor scenario provided in 'O1' of the DeepMIMO dataset [28]. Two cases are considered: (1) $K_t = K_u = 6$, and (2) $K_t = K_u = 12$. In case (1), the BSs 5-8, 15, 16 in Fig. 8 are selected, while BSs 3-10, 15, 16, 17, 18 in Fig. 8 are selected in case (2). UEs are selected randomly on the streets (the black region of Fig. 8). For both cases, BLN-PGP is trained with $N_t = 64$ $(X = 1, Y = Z = 8)$ and the PGP solutions. The performance results for the two cases are shown in Table V(a) and Table V(b), respectively. One can observe that for both cases the proposed BLN-PGP can yield high accuracy and generalizes well when deployed in scenarios with different number of transmit antennas.

In Table. VI, we further consider the generalization capability with respect to the number of BSs K_t. The BLN-PGP is trained under the setting of case (2) with $K_t = 12$, $K_u = 12$ and $N_t = 64$ while is tested in different scenarios with $K_t = 6$, $K_t = 18$ and randomly selected numbers of BSs. It can be observed from the table that the proposed BLN-PGP can maintain good performance and has good generalization capability w.r.t the number of BSs.

VI. Conclusion

In this paper, we have considered a learning-based beamforming design for MISO-ICs and cooperative multicell scenarios. In particular, in order to overcome the computational issues of massive MIMO beamforming optimization, we have proposed the BLN-PGP by unfolding the simple PGP method. We have shown that by exploiting the low-dimensional structures of optimal beamforming solutions and by removing the

dependence of the input/output of the MLP on the number of BSs, the proposed BLN-PGP can have a low complexity, and such complexity is independent of the numbers of transmit antennas and BSs. Extensive experiments based on both synthetic channel and the DeepMIMO dataset have demonstrated that the proposed BLN-PGP can achieve a high solution accuracy with the expense of a small computation time. More importantly, the proposed BLN-PGP has promising generalization capabilities with respect to the number of transmit antennas, the number of BSs, and the cell radius, which is a key ability to be employed in heterogeneous networks. It is worthwhile to point out that the proposed BLN provides a flexible design framework, and it can be extended to other scenarios such as the broadcast interfering channels [13].

REFERENCES

[1] M. Zhu and T.-H. Chang, "Optimization inspired learning network for multiuser robust beamforming," in *IEEE 11th Sensor Array and Multichannel Signal Processing Workshop*, Hangzhou, China, Jun. 8-11, 2020, pp. 1–5.

[2] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, 2014.

[3] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G radio access technology," *IEEE Commun. Stand. Mag.*, vol. 1, no. 4, pp. 24–30, 2017.

[4] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wirel. Commun.*, vol. 23, no. 1, pp. 72–79, 2016.

[5] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 62–75, 2010.

[6] W.-K. Ma, "Semidefinite relaxation of quadratic optimization problems and applications," *IEEE Signal Process. Mag.*, vol. 1053, no. 5888/10, 2010.

[7] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, 2008.

[8] Y.-F. Liu, Y.-H. Dai, and Z.-Q. Luo, "Coordinated beamforming for MISO interference channel: Complexity analysis and efficient algorithms," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1142–1157, 2010.

[9] 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, 2008.

[10] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3711–3724, 2009.

[11] C. Shen, W.-C. Li, and T.-H. Chang, "Wireless information and energy transfer in multi-antenna interference channel," *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6249–6264, 2014.

[12] S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, 2008.

[13] Q. Shi, M. Razaviyayn, Z.-Q. 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, 2011.

[14] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *arXiv preprint arXiv:1707.07980*, 2017.

[15] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun*, vol. 37, no. 6, pp. 1248–1261, 2019.

[16] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, 2018.

[17] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun*, vol. 68, no. 3, pp. 1760–1776, 2019.

[18] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, 2019.

[19] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[20] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *IEEE International Workshop on Signal Processing Systems*, Nanjing, China, Oct. 20-23, 2019, pp. 266–271.

[21] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[22] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, 2019.

[23] M.-W. Un, M. Shao, W.-K. Ma, and P. Ching, "Deep MIMO detection using ADMM unfolding." in *DSW*, 2019, pp. 333–337.

[24] Y. Wei, M.-M. Zhao, M. Hong, M.-J. Zhao, and M. Lei, "Learned conjugate gradient descent network for massive MIMO detection," in *Proc. IEEE ICC*, Dublin, Ireland, Jun. 7-11, 2020, pp. 1–6.

[25] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, 2020.

[26] L. Pellaco, M. Bengtsson, and J. Jaldén, "Deep unfolding of the weighted MMSE beamforming algorithm," *arXiv preprint arXiv:2006.08448*, 2020.

[27] D. Bertsekas, *Nonlinear Programming, 2nd ed.* Athena Scientific Belmont, MA, 1999.

[28] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," *arXiv preprint arXiv:1902.06435*, 2019.

[29] E. G. Larsson and E. A. Jorswieck, "Competition versus cooperation on the MISO interference channel," *IEEE J. Sel. Areas Commun*, vol. 26, no. 7, pp. 1059–1069, 2008.

[30] R. H. Tehrani, S. Vahid, D. Triantafyllopoulou, H. Lee, and K. Moessner, "Licensed spectrum sharing schemes for mobile operators: A survey and outlook," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 4, pp. 2591–2623, 2016.

[31] M. Biguesh and A. B. Gershman, "Training-based mimo channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, 2006.

[32] H. Tuy, "Monotonic optimization: Problems and solution approaches," *SIAM J. Optim*, vol. 11, no. 2, pp. 464–494, 2000.

[33] S. Verdu *et al.*, *Multiuser detection*. Cambridge university press, 1998.

[34] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[35] W. Utschick and J. Brehmer, "Monotonic optimization framework for coordinated beamforming in multicell networks," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1899–1909, 2011.

[36] W.-C. Li, T.-H. Chang, and C.-Y. Chi, "Multicell coordinated beamforming with rate outage constraint—Part II: Efficient approximation algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2763–2778, 2015.

[37] E. A. Jorswieck, E. G. Larsson, and D. Danev, "Complete characterization of the Pareto boundary for the MISO interference channel," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5292–5296, 2008.

[38] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun*, vol. 37, no. 10, pp. 2239–2250, 2019.

[39] W.-K. Ma, J. Pan, A. M.-C. So, and T.-H. Chang, "Unraveling the rank-one solution mystery of robust MISO downlink transmit optimization: A verifiable sufficient condition via a new duality result," *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1909–1924, 2017.

[40] M. K. Karakayali, G. J. Foschini, and R. A. Valenzuela, "Network coordination for spectrally efficient communications in cellular systems," *IEEE Wirel. Commun.*, vol. 13, no. 4, pp. 56–61, 2006.

[41] H. Zhang and H. Dai, "Cochannel interference mitigation and cooperative processing in downlink multicell multiuser MIMO networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2004, no. 2, p. 202654, 2004.

[42] E. Bjornson, R. Zakhour, D. Gesbert, and B. Ottersten, "Cooperative multicell precoding: Rate region characterization and distributed strategies with instantaneous and statistical CSI," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4298–4310, 2010.

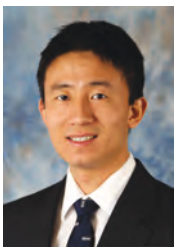
[43] E. LTE, "Evolved Universal Terrestrial Radio Access (E-UTRA); radio frequency (RF) requirements for LTE Pico Node B (3GPP TR 36.931 version 9.0. 0 Release 9)," 2011.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3230662

Minghe Zhu received the B.E. degree in Communication Engineering from Tongji University, Shanghai, China, in 2014 and the M.S. degree in the Communication and Information System from the University of Chinese Academy of Sciences, Shanghai, China, in 2017. He is currently pursuing the Ph.D. degree at the School of Science and Engineering (SSE), The Chinese University of Hong Kong, Shenzhen (CUHK-SZ), China. In 2015, he visited Southern University of Science and Technology. His research interests include machine learning, optimization, and their applications in wireless communications and signal processing.

Tsung-Hui Chang (S'07–M'08-SM'17-F'23) is currently the Associate Professor at the School of Science and Engineering (SSE), The Chinese University of Hong Kong, Shenzhen (CUHK-SZ), China, and Shenzhen Research Institute of Big Data. Previously, he was an Assistant Professor at the National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan (2012-2015), and a Postdoctoral researcher at National Tsing Hua University (NTHU), Hsinchu, Taiwan (2008-2011), and the University of California, Davis, CA, USA (2011-2012). His research interests include signal processing and optimization problems in data communications and machine learning.

Dr. Chang received the Young Scholar Research Award of NTUST in 2014, the IEEE ComSoc Asian-Pacific Outstanding Young Researcher Award in 2015, the IEEE Signal Processing Society (SPS) Best Paper Award in 2018 and 2021, and the Outstanding Faculty Research Award of SSE, CUHK-SZ, in 2021. Dr. Chang is an Elected Member of the IEEE SPS Signal Processing for Communications and Networking Technical Committee (SPCOM TC) (2020/01-) and the Founding Chair of the IEEE SPS Integrated Sensing and Communication Technical Working Group (ISAC TWG). He has served the editorial board for major SP journals, including an Associate Editor (2014/08-2018/12) and Senior Area Editor (2021/02-present) of IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Associate Editor of IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS (2015/01-2018/12) and IEEE OPEN JOURNAL OF SIGNAL PROCESSING (2020/01-present).



Mingyi Hong received his Ph.D. degree from the University of Virginia, Charlottesville, in 2011. He is currently an Associate professor in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. He serves on the IEEE Signal Processing for Communications and Networking Technical Committee, and he is an Associate Editor for IEEE Transactions on Signal Processing. In the past, he has also served on IEEE Machine learning for Signal Processing Technical Committees. His research interests include optimization theory and applications in signal processing and machine learning. His work has received a number of awards, including an IEEE Signal Processing Society Best Paper Award, an International Consortium of Chinese Mathematicians Best Paper Award, and a Best Student Paper Award from NeurIPS workshops. He is also a recipient of an IBM Faculty Award, a Meta Research Award, and he is an Amazon Scholar.