Joint Precoding and CSI Dimensionality Reduction: An Efficient Deep Unfolding Approach

Jianjun Zhang, *Member, IEEE*, Christos Masouros, *Senior Member, IEEE*, and Lajos Hanzo, *Life Fellow, IEEE*

***Abstract*—A recently proposed unified precoding and pilot design optimization (UPPiDO) framework offers a reduction in both training and feedback overhead of acquiring channel state information (CSI) and an enhancement in robustness (to CSI uncertainties) at the expense of a more computationally demanding precoding optimization. To address this increased complexity, in this paper we first propose an unfolding-friendly iterative algorithm, which can efficiently address a family of non-convex and non-smooth problems. Then, we develop an efficient approach to unfold the iterative algorithm designed. Besides being applicable to important and typical iterative optimization algorithms, a pivotal advantage of the proposed unfolding approach is that the trainable parameters are scalars (rather than matrices). This, in turn, reduces the number of training samples required and makes it suitable for rapidly fluctuating wireless environments. We apply the algorithm unfolding (AU) techniques developed to our UPPiDO-based symbol-level precoding and block-level precoding. Our complexity analysis indicates that the computational complexity is scalable both with the numbers of served users and antennas. Our simulation results demonstrate that the number of outer iterations (or layers) required is about 1/3 of that of the original iterative algorithms.**

***Index Terms*—Algorithm unfolding, symbol-level precoding, block-level precoding, MIMO communications, unified precoding and pilot design optimization, complexity reduction.**

I. Introduction

Massive multiple-input multiple-output (MIMO) techniques that employ a large number of antennas at the base station (BS) to simultaneously serve multiple users have received considerable attention in wireless communications, thanks to their high spectrum and energy efficiency, while relying on appealingly low-complexity signal processing techniques [1]. The most fundamental technique of reaping these benefits is to maximize a performance metric of interest via solving an optimization-based transmit precoding problem, which has been an active research area [2]. In the past twenty years, a variety of precoding algorithms have been proposed to improve the system performance [3]–[7], typically with the goal of maximizing the system's sum-rate.

Recently, the symbol-level precoding (SLP), an important branch of precoding design, has received particular attention, thanks to its prominent advantages, involving the constructive utilization of interference for improving the energy efficiency. In contrast to classical precoding methods, where interferences are often regarded as a limitation and are suppressed, interferences in SLP are actively exploited from an instantaneous point of view. In particular, the concept of constructive interference (CI) was exploited to improve system performance in [8]–[12]. Noticeably, a low-complexity vector precoding scheme was proposed in [12] for the limited-feedback multi-user MISO downlink, which is the first work on optimization-based CI precoding. This was followed by [13], proposing explicit precoding optimization for the first time based on CI with strict angle constraints, which was then extended in [14] to a CSI-robust CI precoding relying on relaxed optimization.

Although the iterative precoding algorithms of [6]–[13], [15], [16] achieve optimal performance, their increased transmitter complexity impedes their application in real-time systems, especially for large-scale multi-antenna systems. The reason for this is three-fold. First, commonly-used operations such as the inversion and multiplication of matrices of large dimension are very expensive, which is exacerbated through numerous iterations within an iteration-based precoding algorithm. Second, the number of optimization problems to be solved may be excessive. For example, in SLP it depends on the product of sizes of constellations of all served users. Third, to exploit the specific features of the problems studied, non-smooth regularization or penalty terms are incorporated, which leads to increased computational complexity [17]–[20]. As an example, when UPPiDO [18] is incorporated into precoding, the $L1$ penalty/regularization term is added into the objective function to achieve the goal of pilot selection, which increases the complexity. Thus, efficient algorithms having low complexity are sought for precoding designs.

To address computation-intensive and time-sensitive problems in signal processing and wireless communications, machine learning based algorithms have been developed [21]–[31]. Recently, an insightful taxonomy has been proposed in [32], whose key is to identify the presence of a continuum in terms of specificity and parameterization and view deep learning [21]–[24] and model-based optimization [6]–[13] as two ends of the continuous spectrum. Based on this perspec-

The work was supported by the Engineering and Physical Sciences Research Council, UK under project EP/S028455/1, in part by the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/X01228X/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028), in part by the Scientific Research Fund of the Nanjing University of Aeronautics and Astronautics, and in part by the Research Project of Jiangsu Province under Grant SBK2023045652. (Corresponding author: Lajos Hanzo.)

J. Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (E-mail: jianjun.zhang@nuaa.edu.cn).

C. Masouros is with the Department of Electronic & Electrical Engineering, University College London, London WC1E7JE, U.K. (E-mail: c.masouros@ucl.ac.uk).

L. Hanzo is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO171BJ, U.K. (E-mail: lh@ecs.soton.ac.uk).

tive, hybrid model and deep learning techniques with different levels of specificity and parameterization lie in the middle part of the spectrum. Compared to deep neural network based black-box solutions, which typically require a huge number of training samples and suffer from poor interpretability, the model-based learning techniques can enjoy the advantages of both model-based and learning-aided solutions while overcoming their drawbacks. The hybrid model and (deep) learning techniques fall into three categories [32], namely: 1) coined learned optimizers [33], [34]; 2) deep unfolding, also referred to as algorithm unfolding (AU) [25]–[31]; and 3) deep neural network aided optimizers [35].

The model-based AU approach [25]–[31] has been shown to lead to superior performance, which is thus the focus of this paper. The key of AU unfolds the iterations of an existing iterative algorithm into a neural-network-analogous layer-wise structure and optimizes the relevant hyper-parameters via gradient descent and back-propagation methods [25]. The AU approach accommodates different strategies, and two typical strategies are that hyper-parameters or per-iteration objective parameters are chosen as the trainable parameters [32]. A deep-unfolding algorithm has been recently proposed for the popular weighted minimum mean-square error (WMMSE) iterative precoding optimization algorithm [26]. In terms of symbol detection, a data-driven model-based symbol detector, referred to as ViterbiNet, was proposed in [27].

Although AU is a promising approach to reduce the computational complexity of optimization-based iterative precoding design, only specific AU-based solutions have been developed for specific iterative algorithms (e.g., the classical WMMSE algorithm [26]). An efficient AU approach for general iterative optimization-based precoding algorithms is still unavailable. The reason is that many optimization-based precoding designs (in particular, continuous convex optimization) depend on interior-point methods (IPMs), e.g., the primal-dual interior-point method. However, complicated and/or non-differentiable operations (e.g., to solve systems of linear equations and use line search tricks) within IPMs prevent the use of back-propagation to train a network, which thus hinders the employment of AU in general IPM-based algorithms. Recently, an AU method has been proposed in [38] for the family of proximal-IPM algorithms, which, however, can only tackle very simple constraints, i.e., affine, hyper-slab and bounded l_2 norm constraints. When it comes to the class of algorithms conceived for more sophisticated composite optimization problems (e.g., UPPiDO-based precoding), the issue of AU is even harder.

In this paper, we propose an efficient technique of unfolding an iterative optimization algorithm tailored for fairly general kind of non-convex and/or non-smooth optimization problems, which includes UPPiDO-based precoding designs. The novel contributions of this paper are contrasted to the literature in Table-I. First, we propose an unfolding-friendly iterative optimization algorithm for a family of non-convex and non-smooth problems extensively harnessed both in signal processing and machine learning. The proposed algorithm is eminently suitable for exploiting the specific properties, features and structures of the underlying problems. Then, we unfold the iterative optimization algorithm obtained by conceiving an efficient iterative algorithm for solving the sub-problems (within the iterative algorithm derived) and for treating the salient hyper-parameters as learnable parameters to obtain a deep network. In view of the fact that almost all iterative algorithms expect large step-size parameters at the beginning and hope to decrease them gradually as the iterations proceed [39], [40], the strategy that makes the hyper-parameters learnable yields better generalization and robustness at a reduced sampling complexity. We apply the proposed AU techniques both to UPPiDO-based SLP and to block-level precoding (BLP) problems. The main contributions are summarized as follows:

- To enable and facilitate AU, by exploiting the popular alternating optimization technique and the method of multipliers we propose an efficient iterative optimization algorithm for an important family of non-convex and non-smooth problems (including the UPPiDO-based precoding form as a special case).
- To accelerate convergence, we further unfold the iterative optimization algorithm derived. In particular, we propose to employ the Uzawa's method [39] for solving the sub-problems within the iterative algorithm obtained, based on which a deep network accrues quite naturally.
- In addition to reducing the training and feedback overhead of acquiring CSI, we highlight that the UPPiDO-based precoding method also helps to enhance robustness to CSI uncertainties. For the UPPiDO-based SLP/BLP problems, we propose efficient iterative algorithms.
- Because the optimization objective of the UPPiDO-based design involves a non-smooth term, which leads to increased complexity, we harness the proposed AU techniques for unfolding the iterative algorithms designed, so as to achieve the desired complexity reduction.

Both our complexity analysis and our comprehensive simulation results confirm the superiority of the proposed algorithms. In particular, the simulation results demonstrate that improved robustness can be achieved at a much-reduced complexity when incorporating both UPPiDO and AU techniques. In terms of the complexity reduction, the number of iterations required by our AU-based algorithms is about $1/3$ of that of the original iterative optimization algorithms.

The remainder of this paper is organized as follows. The system models of the precoding designs with and without UPPiDO are described in Section II and Section III, respectively. Efficient iteration-based optimization algorithm and deep-unfolding based solution are proposed for a set of challenging problems in Section IV. Iterative algorithms and AU-based solutions are proposed for SLP and BLP in Section V and Section VI, respectively. The simulation results are provided in Section VII, and the conclusion is offered in Section VIII.

Throughout the paper, we use the following notations. Bold uppercase $\mathbf{A}$ and bold lowercase $\mathbf{a}$ denote matrices and column vectors, respectively. Non-bold letters A, a represent scalars. Caligraphic letters, such as $\mathcal{A}$, stand for sets. $\mathbb{E}(\cdot)$ and $(\cdot)^{\mathrm{H}}$ denote the mathematical expectation and Hermitian operators, respectively. $\mathbb{I}\{\cdot\}$ and $\mathrm{card}(\mathcal{A})$ are the indicator function and the cardinality of $\mathcal{A}$, respectively. The notation $(\cdot)^{\star}$ denotes an

Representative Algorithm	[21]	[26]	[36]	[37]	This Work
Learnable Parameters	Matrix	Matrix	Scalar	Matrix	Scalar
Sample Complexity	High	High	Moderate	Moderate	Low
CSI Training/Feedback Overhead	Large	Large	Large	Scalable	Scalable
Training Complexity	High	High	Moderate	Moderate	Low
Factors Improving Robustness	✗	✗	CI	CI	Joint CI and CSI Reduction
Solvable Problem (via AU)	✗	Specific	Specific	Specific	Fairly General
Type of Solvable Problem (via AU)	✗	Smooth	Convex	Convex	(Even) Non-smooth Non-convex

TABLE I
COMPARISONS OF REPRESENTATIVE LEARNING-BASED PRECODING ALGORITHMS

optimal quantity, e.g., an optimal solution. $\mathcal{CN}(\mathbf{m},\mathbf{R})$ stands for a complex Gaussian random vector with mean $\mathbf{m}$ and covariance matrix $\mathbf{R}$. The notation $\succeq$ represents component-wise greater than or equal operation.

II. System Model

Consider the FDD downlink of a single BS equipped with N transmit antennas for supporting U single-antenna users. The set of U users is denoted by $\mathcal{U}=\{1,\cdots,U\}$. The users are randomly and uniformly distributed in the coverage area of the BS. A uniform linear array (ULA) is considered. It is relatively straightforward to extend our designs to other types of antenna array geometries (e.g., uniform planar array).

Under the assumption of ULA, the channel vector between the BS and each user u can be expressed as [41], [42]

$$\begin{aligned}\bar{\mathbf{h}}_u &= \int_{\theta\in\mathcal{A}_u} g_u(\theta)\mathbf{a}(\theta)d\theta \\ &= \int_{\bar{\theta}_u-\Delta_u}^{\bar{\theta}_u+\Delta_u} |g_u(\theta)|e^{j\psi_u(\theta)}\mathbf{a}(\theta)d\theta. \end{aligned} \tag{1}$$

Here, $\mathcal{A}_u=[\bar{\theta}_u-\Delta_u,\bar{\theta}_u+\Delta_u]$ represents the angular spread of user u with Δ_u being the single-sided angular spread, $|g_u(\theta)|$ and $\psi_u(\theta)$ represent the attenuation (amplitude) and random phase of the signal ray from θ, respectively, and $\mathbf{a}(\cdot)$ represents the array response vector

$$\mathbf{a}(x)=\frac{1}{\sqrt{N}}\left[1,e^{jx\frac{2\pi}{\lambda}d},e^{jx\frac{2\pi}{\lambda}2d},\cdots,e^{jx\frac{2\pi}{\lambda}(N-1)d}\right],$$

where λ and d represent the signal wave-length and distance between any two adjacent antennas, respectively. We consider a pair of typical precoding schemes, namely the CI-constraint based SLP and sum-rate maximization based BLP.

A. CI-Constraint based Symbol-Level Precoding

For notational simplicity, PSK modulation (with constellation $\mathcal{S}_u$ of size B_u for user u) is considered here. But, the algorithms developed are also applicable to other modulations [11]. Let $s_u=e^{j\xi_u}\in\mathcal{S}_u$ be the intended information symbol for user u (with ξ_u the argument of s_u) and $\mathbf{v}$ be the transmitted signal. The signal received at each user u can be written as

$$y_u=\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}+w_u,$$

where $w_u\sim\mathcal{CN}(0,\sigma^2)$ denotes random noise.

To improve energy efficiency, the idea of CI is exploited. For PSK modulation, the key of the CI design principle can be captured by the following constraints ($\forall\, u\in\mathcal{U}$) [14]

$$\left|\mathrm{Im}(\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}e^{-j\xi_u})\right|\le\left(\mathrm{Re}(\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}e^{-j\xi_u})-\gamma_u\right)\tan(\pi/K_u), \tag{2}$$

where γ_u quantifies the quality of received signal (QoRS) of user u. The above design constraints enforce that the CI pushes the received signal away from the decision boundaries of the PSK constellation. The interested reader is referred to [14] for more details and [43] for an extensive overview of SLP and its application to other modulation formats. As an example, we consider the power-minimization SLP problem, which can be formulated as [14]

$$\begin{aligned}\min_{\mathbf{v}}\quad & \|\mathbf{v}\|^2 \\ \text{s.t.}\quad & \left|\mathrm{Im}(\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}e^{-j\xi_u})\right|\le\left(\mathrm{Re}(\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}e^{-j\xi_u})-\gamma_u\right)\cdot \\ & \qquad \tan(\pi/K_u),\ (\forall\, u\in\mathcal{U}).\end{aligned} \tag{3}$$

B. Sum-Rate Maximization based Block-Level Precoding

To demonstrate the generality of our AU approach, we further consider a classical block-level precoding design. Let $\mathbf{v}_u\in\mathbb{C}^{N\times 1}$ and $s_u\sim\mathcal{CN}(0,1)$ represent the precoding vector and the data stream of user $u\in\mathcal{U}$, respectively. Then, the signal received by user $u\in\mathcal{U}$ is expressed as

$$y_u=\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_u s_u+\sum_{v\neq u}\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_v s_v+w_u.$$

The signal to interference-plus-noise ratio (SINR) γ_u of the user $u\in\mathcal{U}$ is calculated as

$$\gamma_u=\frac{\left|\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_u\right|^2}{\sum_{v\neq u}\left|\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_v\right|^2+\sigma^2}.$$

The optimization problem of the classical sum-rate maximization based BLP is formulated as

$$\begin{aligned}\max_{\{\mathbf{v}_u\}}\quad & \sum_{u\in\mathcal{U}}\log\left(1+\gamma_u\right) \\ \text{s.t.}\quad & \sum_{u\in\mathcal{U}}\|\mathbf{v}_u\|^2\le P_{\mathrm{b}},\end{aligned} \tag{4}$$

where P_{b} denotes the power budget of the BS.

Note that solving Problems (3) and (4) requires physical CSI (pCSI) $\{\bar{\mathbf{h}}_u\}$. However, it is difficult to obtain high-precision pCSI for a FDD system, since the reciprocity between the uplink and downlink channels cannot be exploited. Hence, the

overhead of downlink training and uplink feedback may be prohibitively high. Furthermore, the CSI feedback also suffers from quantization noise and channel errors. To tackle this issue, instead of pCSI, the UPPiDO is proposed to design the precoders using modified CSI (mCSI), defined as $\{\mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u\}$ with $\mathbf{F}$ a matrix of appropriate dimension [18]. Next, we briefly introduce UPPiDO for completeness.

III. Principle, Problem Reformulation and Extension of Unified Precoding and Pilot Design Optimization (UPPiDO)

In this section, we elaborate on applying UPPiDO to the above pair of typical precoding designs. Due to space limitation, the principle of UPPiDO can only be briefly introduced here. We refer interested readers to [18] for details.

A. A Brief Introduction of UPPiDO

The UPPiDO framework incorporates two key support techniques, i.e., mCSI based precoding and pilot design (mCSI-PPD) as well as mCSI based intelligent sensing and selection (mCSI-ISS). On the one hand, the key to reducing the overhead of training and feedback is to exploit the channel's sparsity. To this end, mCSI-PPD jointly optimizes the pilot and precoding in another domain, where the corresponding mCSI is sparse. On the other hand, even though the mCSI is sparse, it may still contain redundant and/or inaccurate information. Our mCSI-ISS technique tackles this issue by identifying and selecting significant mCSI. Specifically, automatic model selection via $L1$ regularization (or other sparsity-induced regularizations) is utilized to induce, identify and select significant mCSI.

Let $\mathbf{v}_u \in \mathbb{C}^N$ denote the precoding vector of user u. The set $\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_w\}$ collects all terms that take the form of the inner product between the channel vectors and precoding vectors, i.e.,

$$\big\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_w\big\} = \big\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_w \,|\, u \in \mathcal{U}, w \in \mathcal{U}\big\}. \tag{5}$$

The problem of a typical precoding design (including Problems (3) and (4) as special cases) can be formulated as [1]

$$\begin{aligned} \min_{\{\mathbf{v}_w\}} \quad & f\big(\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_w\}\big) \\ \text{s.t.} \quad & g_i\big(\{\bar{\mathbf{h}}_u^{\mathrm{H}}\mathbf{v}_w\}\big) \le 0,\ (i \in \mathcal{I} = \{1, \cdots, I\}), \end{aligned} \tag{6}$$

where $\{g_i(\cdot) \le 0\}$ represent inequality constraints. Note that in Problem (6), the physical channel vectors $\{\bar{\mathbf{h}}_u\}$ never emerge alone. Instead, they always take the form of an inner product with the precoding vectors. Since the only requirement is that the optimization variables of interest and $\{\bar{\mathbf{h}}_u\}$ satisfy the form of inner product, the form in (6) is very general and contains many precoding designs of interest.

The theoretical foundation of the mCSI-PPD technique is provided in Theorem 1.

[1]Because an equality constraint $e(\mathbf{x}) = 0$ is equivalent to two inequality constraints (i.e., $e(\mathbf{x}) \le 0$ and $e(\mathbf{x}) \ge 0$), problem (6) has contained equality constraints. But the form in (6) simplifies the description of our approach.

Theorem 1 ([18])**.** *Let $\mathbf{F}$ be a matrix such that the set of all column vectors of $\mathbf{F}$, denoted by $\mathcal{F}$, spans the vector space $\mathbb{C}^N$. Then, Problem* (6) *is equivalent to the following problem*

$$\begin{aligned} \min_{\{\mathbf{y}_w\}} \quad & f\big(\{\bar{\mathbf{h}}_u^{H}\mathbf{F}\mathbf{y}_w\}\big) \\ \textit{s.t.} \quad & g_i\big(\{\bar{\mathbf{h}}_u^{H}\mathbf{F}\mathbf{y}_w\}\big) \le 0,\ (\forall\, i \in \mathcal{I}). \end{aligned} \tag{7}$$

Moreover, let the set $\{\mathbf{y}_w^\star\}$ be an optimal solution of Problem (7)*. Then, the set $\{\mathbf{v}_w^\star = \mathbf{F}\mathbf{y}_w^\star\}$ is an optimal solution of Problem* (6)*.*

Theorem 1 indicates that to solve Problem (6), it is sufficient to solve the equivalent optimization problem in (7). Compared to the original problem in (6), the advantage of the equivalent problem in (7) is that there is no need to estimate pCSI $\{\bar{\mathbf{h}}_u\}$. [2] Instead, if the mCSI, defined as $\{\mathbf{h}_u = \mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u\}$, is available, an efficient precoder can still be obtained. More importantly, by appropriately designing $\mathbf{F}$ (or $\mathcal{F}$), the acquisition of $\{\mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u\}$ may be much easier, e.g., lower training overhead. For example, via appropriate design, $\{\mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u\}$ may be sparse, and thus the CSI acquisition results in low feedback.

Remark 3.1 To obtain the optimal performance, $\mathbf{F}$ should also be optimized, which is, however, beyond the scope of this paper. An efficient approach has been proposed in [37]. In view of channel model (1) (e.g., with limited angular spread), $\mathbf{F}$ is chosen to be the DFT codebook in Section VII.

As $\mathcal{F}$ plays the role of estimating mCSI, it can be reasonably referred to as a pilot. However, $\mathcal{F}$ may provide redundant or useless information and thus it is inefficient. The key to reduce training and feedback overhead is to carefully choose a subset from $\mathcal{F}$, which is defined as intelligent pilot (IP) and denoted by $\mathcal{F}_{\mathrm{IP}}$. However, it is non-trivial to choose an optimal IP. The mCSI-ISS addresses this issue by exploiting the property that the $L1$ regularizer tends to induce a sparse solution, which allows us to automatically identify and choose an IP. Then, the problem in (7) can be further reformulated as

$$\begin{aligned} \min_{\{\mathbf{y}_w\}} \quad & f\big(\{\mathbf{h}_u^{\mathrm{H}}\mathbf{y}_w\}\big) + \rho\sum_w \|\mathbf{y}_w\|_1 \\ \text{s.t.} \quad & g_i\big(\{\mathbf{h}_u^{\mathrm{H}}\mathbf{y}_w\}\big) \le 0,\ (\forall\, i \in \mathcal{I}), \end{aligned} \tag{8}$$

where $\rho > 0$ is a regularization constant. Because of the term $\rho\sum_w \|\mathbf{y}_w\|_1$, the optimal solution of Problem (8) is often sparse, based on which an optimal IP can be identified.

B. UPPiDO-based Problem Reformulation

Based on the above discussion, it can be verified that the SLP problem in (3) can be reformulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{F}\mathbf{x}\|^2 + \rho\|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \big|\mathrm{Im}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u})\big| \le \Big(\mathrm{Re}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u}) - \gamma_u\Big)\cdot \\ & \qquad \tan(\pi/K_u),\ (\forall\, u \in \mathcal{U}). \end{aligned} \tag{9}$$

[2]By regarding each column of $\mathbf{F}$ (or each element of $\mathcal{F}$) as a transmit beam (with transmit symbol $s = 1$), the signal received at user u is given by

$$\mathbf{y}_u = \sqrt{p}\mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u s + \mathbf{w},$$

where p denotes transmit power and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ is the received noise vector. Then, $\mathbf{y}_u/\sqrt{p}$ is, in fact, an effective estimate of $\mathbf{h}_u$.

Similarly, the BLP problem in (4) can be reformulated as

$$\begin{aligned}\max_{\{\mathbf{x}_u\}} \quad & \sum_{u\in\mathcal{U}} \log\left(1+\frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\right|^2}{\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2+\sigma^2}\right)-\rho\sum_{u\in\mathcal{U}}\|\mathbf{x}_u\|_1\\ \text{s.t.} \quad & \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2\leq P_{\mathrm{b}}.\end{aligned} \tag{10}$$

Remark 3.2 Besides the advantage of reducing training and feedback overhead, another important advantage of UPPiDO-based precoding design in (9) (or (10)) is that compared to the original form in (3) (or (4)) and the mCSI-based problem formulation, more robust performance can be achieved for inaccurate pCSI or mCSI. Numerical results are provided in Section VII to demonstrate this point, while the theoretical analysis and proof are deferred to the future work.

Remark 3.3 Since Problem (9) (or (10)) is degenerated into Problem (3) (or (4)) if $\mathbf{F}=\mathbf{I}$ and $\rho=0$, it is sufficient to consider the more general Problem in (9) (or (10)). In fact, all algorithms designed for Problem (9) (or (10)) trivially apply to the specific problem in (3) (or (4)).

Although many appealing advantages can be achieved via the UPPiDO-based composite optimization in (8), the reformulated problems become much more complex than the original problems. In particular, because the $L1$ regularization term $\rho\|\mathbf{x}\|_1$ (or $\lambda\sum_{u\in\mathcal{U}}\|\mathbf{x}_u\|_1$) is non-differentiable, Problem (9) (or (10)) is non-smooth [40], which lowers the convergence rate and thus increases the overall computational complexity. In the next section, we will first develop an efficient iterative optimization algorithm, and then unfold it for further reducing the computational complexity.

For better describing the algorithm that solves Problem (8) later, we compactly write it as

$$\begin{aligned}\min_{\mathbf{x}} \quad & f(\mathbf{x})+\rho R(\mathbf{x})\\ \text{s.t.} \quad & g_i(\mathbf{x})\leq 0,\ (\forall\, i\in\mathcal{I}),\end{aligned} \tag{11}$$

where $\mathbf{x}$ denotes the optimization variable of appropriate dimension. We further stipulate the following assumptions:

- Each $g_i(\cdot)$ can be written as the difference of two convex functions, i.e., $g_i(\cdot)=c_i(\cdot)-d_i(\cdot)$ with c_i and d_i both convex (d_i may be zero function, i.e., $d_i=0$). In fact, $g_i(\cdot)$ is often second-order differentiable in many wireless communication designs, which guarantees that it can be written as the difference of two convex functions [44].
- $f(\cdot)$ is differentiable and convex. Note that the assumption of convexity is not restrictive. In fact, if f is not convex, the objective function can be replaced by $t+\rho R(\mathbf{x})$ upon introducing the variable t and constraint $f(\mathbf{x})\leq t$. Then, the new problem satisfies form (11).
- $R(\cdot)$ is convex but non-differentiable. In most cases, $R(\cdot)$ is introduced for capturing the specific structures of the underlying problems. As an example, the $L1$ norm (i.e., $R(\cdot)=\|\cdot\|_1$) is used to induce a sparse solution and select significant variables within a UPPiDO-based solution.

Under the above assumptions, Problem (11) is rewritten as

$$\begin{aligned}\min_{\mathbf{x}} \quad & f(\mathbf{x})+\rho R(\mathbf{x})\\ \text{s.t.} \quad & c_i(\mathbf{x})-d_i(\mathbf{x})\leq 0,\ (\forall\, i\in\mathcal{I}).\end{aligned} \tag{12}$$

Next, we concentrate on the optimization problem in (12).

IV. Iterative Algorithm Design and Efficient Unfolding

In this section, we first propose an unfolding-friendly iterative algorithm. Then, we further unfold the iterative algorithm designed, so as to reduce the computational complexity.

A. Unfolding-friendly Iterative Algorithm

To obtain an unfolding-friendly iterative algorithm, it is of pivotal importance to maintain the underlying structure of $R(\cdot)$. For this purpose, we first introduce an auxiliary variable $\mathbf{z}$ and equivalently write Problem (12) as

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x})+\rho R(\mathbf{z})\\ \text{s.t.} \quad & c_i(\mathbf{x})-d_i(\mathbf{x})\leq 0,\ (\forall\, i\in\mathcal{I})\\ & \mathbf{x}-\mathbf{z}=\mathbf{0}.\end{aligned} \tag{13}$$

To maintain and exploit the structure of R, we further employ the popular alternative optimization method to address Problem (13). Specifically, by introducing the penalty parameter κ and dual variable $\mathbf{y}$ for the constraint $\mathbf{x}-\mathbf{z}=\mathbf{0}$, Problem (13) can be written as

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x})+\rho R(\mathbf{z})+\mathbf{y}^{\mathrm{T}}(\mathbf{x}-\mathbf{z})+\kappa\|\mathbf{x}-\mathbf{z}\|^2/2\\ \text{s.t.} \quad & c_i(\mathbf{x})-d_i(\mathbf{x})\leq 0,\ (\forall\, i\in\mathcal{I}).\end{aligned} \tag{14}$$

Note, however, that the non-convex constraints in (14) still prevent an efficient solution. To tackle this issue, we employ the constrained convex concave procedure [44] to tackle these constraints. Let $\mathbf{x}_n$ represent the n-th iteration of $\mathbf{x}$, the $(n+1)$-st iteration of $\mathbf{x}$ can be obtained by solving

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x})+\rho R(\mathbf{z})+\mathbf{y}^{\mathrm{T}}(\mathbf{x}-\mathbf{z})+\kappa\|\mathbf{x}-\mathbf{z}\|^2/2\\ \text{s.t.} \quad & c_i(\mathbf{x})-d_i(\mathbf{x}_n)-\nabla d_i(\mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_n)\leq 0,\ (\forall\, i\in\mathcal{I}).\end{aligned} \tag{15}$$

To solve Problem (15), it is sufficient to solve the following two sub-problems alternately and update dual variable $\mathbf{y}$.

1) The problem with respect to (w.r.t.) $\mathbf{x}$: The variable $\mathbf{x}$ is updated by solving the following problem

$$\begin{aligned}\min_{\mathbf{x}} \quad & f(\mathbf{x})+\mathbf{y}^{\mathrm{T}}(\mathbf{x}-\mathbf{z})+\kappa\|\mathbf{x}-\mathbf{z}\|^2/2\\ \text{s.t.} \quad & c_i(\mathbf{x})-d_i(\mathbf{x}_n)-\nabla d_i(\mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_n)\leq 0,\ (\forall\, i\in\mathcal{I}).\end{aligned} \tag{16}$$

Since Problem (16) is, in fact, convex, it can be solved via a convex optimization toolbox.

2) The problem w.r.t. $\mathbf{z}$: The auxiliary variable $\mathbf{z}$ can be updated by solving the following optimization problem

$$\min_{\mathbf{z}} \quad \rho R(\mathbf{z})+\mathbf{y}^{\mathrm{T}}(\mathbf{x}-\mathbf{z})+\kappa\|\mathbf{x}-\mathbf{z}\|^2/2. \tag{17}$$

Via algebraic operations, it can be verified that Problem (17) can be equivalently written as

$$\min_{\mathbf{z}} \quad \rho R(\mathbf{z})+\frac{\kappa}{2}\|\mathbf{x}-\mathbf{z}+\mathbf{y}/\kappa\|^2. \tag{18}$$

The optimal solution of Problem (18) can be represented by $\mathrm{prox}_{\rho/\kappa R}(\mathbf{x}+\mathbf{y}/\kappa)$, where $\mathrm{prox}(\cdot)$ is the proximal operator. For a convex function h, its proximal operator is defined as

$$\mathrm{prox}_h(\mathbf{x}) = \underset{\mathbf{u}\in\mathrm{dom}h}{\arg\min}\left\{h(\mathbf{u}) + \frac{1}{2}\|\mathbf{u}-\mathbf{x}\|^2\right\}.$$

In some cases, an analytic expression of $\mathrm{prox}_{\rho/\kappa R}(\mathbf{x}+\mathbf{y}/\kappa)$ is available. In particular, for the $L1$ norm used in this paper (i.e., $R(\mathbf{z}) = \|\mathbf{z}\|_1$), $\mathrm{prox}_{\rho/\kappa R}(\mathbf{x}+\mathbf{y}/\kappa)$ is calculated as

$$\mathbf{z}^\star = \mathrm{sign}(\mathbf{x}+\mathbf{y}/\kappa) \odot \max\{|\mathbf{x}+\mathbf{y}/\kappa| - \rho/\kappa, 0\}, \quad (19)$$

where $\mathrm{sign}(\cdot)$ is the sign function (i.e., $\mathrm{sign}(z) = z/|z|$ for $z \neq 0$) [3], and $\odot$ represents the Hardmard product (i.e., element-wise product) of two vectors. Note that the operations $\mathrm{sign}(\cdot)$, $\max\{\cdot\}$ and $|\cdot|$ in (19) act on their arguments in an element-wise manner. The benefit is that thanks to the element-wise representation, the update in (19) can be operated in parallel.

Finally, we consider the update of the dual variable $\mathbf{y}$. In the n-th iteration, the dual variable $\mathbf{y}$ is updated as

$$\mathbf{y}_n = \mathbf{y}_{n-1} + \kappa(\mathbf{x}_n - \mathbf{z}_n), \quad (20)$$

where $\mathbf{z}_n$ represents the n-th iteration of $\mathbf{z}$. To obtain better convergence performance, the penalty parameter κ should also be updated. The update expression can be chosen as

$$\kappa_{n+1} = \begin{cases} \beta\kappa_n & \|\mathbf{x}_n - \mathbf{z}_n\|^2 > \gamma\|\mathbf{x}_{n-1} - \mathbf{z}_{n-1}\|^2 \\ \kappa_n & \text{otherwise,} \end{cases} \quad (21)$$

where $\beta > 1$ and $\gamma \in (0,1)$ are real numbers introduced to control the increasing rate of κ.

Algorithm 1: Iterative Algorithm for Problem (11) or (12)

1: **initialize:** dual variable $\mathbf{y}$, auxiliary variable $\mathbf{z}$, β and γ

2: **repeat**

(a) **update** primal variable $\mathbf{x}$ by solving (16)

(b) **update** auxiliary variable $\mathbf{z}$ by solving (18)

(c) **update** dual variable $\mathbf{y}$ according to (20)

(d) **update** penalty parameter κ according to (21)

3: **until** some convergence criterion is met

4: **output:** optimal solution $\mathbf{x}^\star$

The complete iteration procedure is summarized in Algorithm 1. The dual variable $\mathbf{y}$ and auxiliary variable $\mathbf{z}$ are first initialized. In Step (a) and Step (b), $\mathbf{x}$ and $\mathbf{z}$ are updated by solving Problem (16) and Problem (18), respectively. The dual variable $\mathbf{y}$ is updated according to (20) in Step (c), and the penalty parameter κ is updated in Step (d) as per (21). The convergence criterion can be $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| \leq \varepsilon$ and $\|\mathbf{z}_n - \mathbf{z}_{n-1}\| \leq \varepsilon$, where $\varepsilon\,(>0)$ denotes a small real number. Although the derivation of the proposed algorithm bears some similarity to ADMM (e.g., [45]), which is mainly applicable to convex problems, our algorithm is also applicable to more challenging non-convex problems. This is attributed to the comprehensive use of sophisticated techniques (e.g., splitting, alternative optimization, local convexification and dual ascent). Thanks to these techniques, compared to the conventional augmented Lagrangian method, our algorithm can fully exploit problem structures (e.g., block or sparsity structure) and has low complexity. Typically, closed-form solutions are available and/or smooth optimization methods (with faster convergence rate) can be applied for the sub-problems.

Note that the performance of the composite optimization algorithm conceived heavily depends on the preassigned hyper-parameters (e.g., the penalty parameter κ and step-size parameters used by the algorithm to solve Problem (16)). However, it is a challenging task to choose them optimally. In fact, they are often chosen heuristically. Moreover, computation-intensive operations (e.g., matrix inversion) within an IPM when solving (16) further exacerbate the issue of escalating computational complexity. Next, we employ the AU techniques to address these challenging issues for enhancing the performance of the iterative algorithm constructed.

B. Network Architecture for Efficient Unfolding

The main difficulty of unfolding Algorithm 1 lies in solving Problem (16). Although Problem (16) can be solved by IPM, many complex operations caused by IPM hinder efficient unfolding. In a nutshell, it is not the best option to unfold an IPM-based iterative algorithm. Here, we tackle this issue by developing an unfolding-oriented iterative algorithm. In view that Problem (16) is convex and the strong duality theorem holds (under some constraint qualification, e.g., Slater's constraint qualification [40]), dual methods can be utilized to address this problem. The Uzawa's method [39] is chosen to solve Problem (16), which can be equivalently written as

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) + \kappa\|\mathbf{x}-\mathbf{z}+\mathbf{y}/\kappa\|^2/2 \\ \text{s.t.} \quad & c_i(\mathbf{x}) - d_i(\mathbf{x}_n) - \nabla d_i(\mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_n) \leq 0,\ (\forall\, i \in \mathcal{I}). \end{aligned} \quad (22)$$

For convenience, let $J(\mathbf{x}) = f(\mathbf{x}) + \kappa\|\mathbf{x}-\mathbf{z}+\mathbf{y}/\kappa\|^2/2$ and $\phi_i(\mathbf{x}) = c_i(\mathbf{x}) - d_i(\mathbf{x}_n) - \nabla d_i(\mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_n)$. The Lagrange dual function of Problem (22) is given by

$$G(\boldsymbol{\mu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}), \quad (\boldsymbol{\mu} \in \mathbb{R}_+^I, I = \mathrm{card}(\mathcal{I})), \quad (23)$$

where $\boldsymbol{\mu}$ denotes the dual variable and $L(\mathbf{x}, \boldsymbol{\mu})$ is the Lagrangian of Problem (22), i.e.,

$$L(\mathbf{x}, \boldsymbol{\mu}) = J(\mathbf{x}) + \sum_{i\in\mathcal{I}} \mu_i \phi_i(\mathbf{x}). \quad (24)$$

The projection-gradient method is applicable to the following dual problem due to its convexity

$$\min_{\boldsymbol{\mu}\in\mathbb{R}_+^I} \quad G(\boldsymbol{\mu}). \quad (25)$$

Uzawa's method [39] is essentially the gradient method with fixed step-size applied to the dual problem in (25), which is also designed to yield a solution of the primal Problem (22). Specifically, given an arbitrary initial dual vector $\boldsymbol{\mu}_1 \in \mathbb{R}_+^I$, the sequences $\{\boldsymbol{\mu}_m\}$ and $\{\mathbf{x}_m\}$ are constructed, and with $\boldsymbol{\mu}_m$ available, $\mathbf{x}_m$ and $\boldsymbol{\mu}_{m+1}$ are determined as follows:

$$\mathbf{x}_m = \underset{\mathbf{x}}{\arg\min}\, L(\mathbf{x}, \boldsymbol{\mu}_m) \quad (26)$$

$$\mu_{m+1,i} = \max\left\{\mu_{m,i} + \alpha\phi_i(\mathbf{x}_m), 0\right\},\ (i \in \mathcal{I}), \quad (27)$$

[3]The real domain is considered here. For the complex domain, $\mathrm{sign}(\cdot)$ is replaced by the complex sign function $\mathrm{sgn}(\cdot)$, i.e., $\mathrm{sgn}(z) = z/|z|$ for $z \neq 0$.

where α denotes the step-size parameter (for the dual update) and $\mu_{m,i}$ represents the i-th component of $\boldsymbol{\mu}$ in the m-th iteration. The procedure of solving Problem (22) is provided in Algorithm 2. Similarly, the convergence criterion is that the duality gap is sufficiently small.

Algorithm 2: Iterative Algorithm for Problem (22)

1: **initialize** dual variable $\boldsymbol{\mu}_1 \in \mathbb{R}_+^I$ (arbitrarily), let $m = 1$

2: **repeat**

(a) **update** primal variable $\mathbf{x}$ by minimizing $L(\mathbf{x}, \boldsymbol{\mu}_m)$

(b) **update** dual variable $\boldsymbol{\mu}$ according to (27)

(c) **update** counter $m \leftarrow m + 1$

3: **until** some convergence criterion is met

4: **output:** optimal (primal) solution $\mathbf{x}_n^\star$

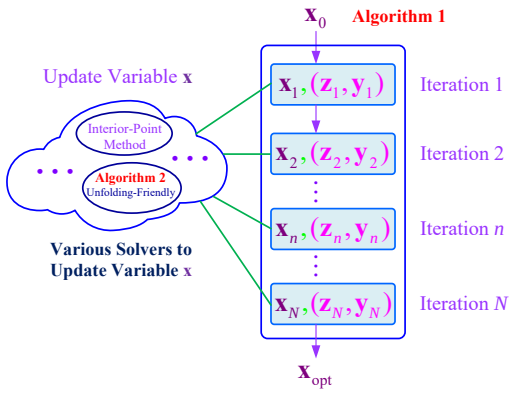

Fig. 1. The relationship between Algorithm 1 and Algorithm 2.

For clarity, the relationship between Algorithms 1 and 2 is illustrated in Fig. 1. As a general algorithm or framework, Algorithm 1 accommodates various methods to update $\mathbf{x}$, among which Algorithm 2 is a particularly unfolding-friendly one. It is also observed that we should first unfold Algorithm 2 (before unfolding Algorithm 1), because it is the basis of unfolding the more complex Algorithm 1. It should be noted that unfolding Algorithm 2 itself is very important, e.g., when the regularization term $\rho R(\cdot)$ is absent. Since the update of $\boldsymbol{\mu}$ in (27) is trivial, it is sufficient to consider (26). The problem of minimizing $L(\mathbf{x}, \boldsymbol{\mu}_m)$ is an unconstrained optimization problem. In some cases, a closed-form solution may be available. So, we consider the case where a closed-form solution is unavailable. In this case, gradient descent can be used to address this problem:

$$\mathbf{x}_{m,l+1} = \mathbf{x}_{m,l} - \eta_m \nabla_{\mathbf{x}} L(\mathbf{x}_{m,l}, \boldsymbol{\mu}_m), \tag{28}$$

where η_m is the step-size within the m-th iteration.

Because parameters α in (27) (for dual update) and η in (28) (for primal update) play a key role, they are chosen as the trainable parameters. In (27), parameter α is shared by all components of $\boldsymbol{\mu}$. To obtain better performance, different components may be updated by different step-sizes. To solve the problem $\min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}_m)$, the gradient descent operation in (28) is repeated B times (i.e., $l = 1, \cdots, B$) for each $\boldsymbol{\mu}_m$, with different step-sizes (i.e., to use step-size $\eta_{m,l}$ for the l-th sub-iteration within the m-th iteration). The updates in (26) and (27) are performed M times (i.e., $m = 1, \cdots, M$). Then, all trainable parameters are collected in set $\mathcal{A}$, i.e.,

$$\begin{aligned}\mathcal{A} = \{\alpha_{m,i}, \eta_{m,l} \,|\, & m = 1, \cdots, M, \\ & i = 1, \cdots, I, l = 1, \cdots, B\}.\end{aligned} \tag{29}$$

The size of $\mathcal{A}$ is $MI + MB$. Given $\mathcal{A}$ and update formulas in (27) and (28), a deep network via AU can be established immediately, whose structure is shown in Fig. 2. Since this network is designed to solve the primal problem in (16), it is referred to as primal problem solving network (PPSN), which is essentially a mapping denoted by $\mathcal{P}(\cdot, \mathcal{A})$.

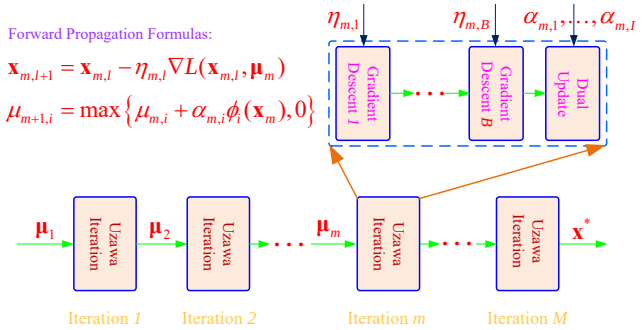

Fig. 2. The structure of the primal problem solving network (PPSN).

Based on PPSN, we can unfold the composite optimization algorithm (i.e., Algorithm 1). For clarity, the structure of the composite optimization algorithm induced deep-unfolding network (COA-DUN) is provided in Fig. 3. COA-DUN consists of K layers (i.e., outer iterations), and each layer consists of three subnetworks/components, i.e., PPSN subnetworks, auxiliary variable update and dual variable update. The trainable parameters of COA-DUN are summarized below:

- $\{\mathcal{A}_n \,|\, n = 1, \cdots, K\}$: $\mathcal{A}_n$ contains all learnable parameters of the n-th PPSN subnetwork.
- $\{\kappa_n \,|\, n = 1, \cdots, K-1\}$: κ_n denotes the penalty parameter corresponding to the n-th layer of COA-DUN.

All trainable parameters are collected in $\boldsymbol{\Theta} = \{\mathcal{A}_n, \kappa_n\}$. The mapping induced by COA-DUN is denoted by $\mathcal{G}(\cdot, \boldsymbol{\Theta})$.

C. Efficient Training Procedure

Now, we discuss how to train COA-DUN. Given a training dataset $\mathcal{D} = \{(\mathcal{X}_1, \mathcal{Y}_1), (\mathcal{X}_2, \mathcal{Y}_2), \cdots, (\mathcal{X}_T, \mathcal{Y}_T)\}$ [4], the trainable parameters are learned as follows. For each sample $(\mathcal{X}_j, \mathcal{Y}_j)$, the forward propagation yields a prediction output $\mathcal{G}(\mathcal{X}_j, \boldsymbol{\Theta})$. Then, the training loss is calculated as

$$L(\mathcal{X}_j, \mathcal{Y}_j) = \frac{1}{2} \left\| \mathcal{G}(\mathcal{X}_j, \boldsymbol{\Theta}) - \mathcal{Y}_j \right\|^2. \tag{30}$$

Since the loss is available, the trainable parameters can be updated via the back-propagation method. Because all mathematical operations within the mapping $\mathcal{G}(\cdot, \boldsymbol{\Theta})$, e.g., (27) and (28), are differentiable or differentiable almost everywhere, the

[4]The input $\mathcal{X}$ and label $\mathcal{Y}$ of COA-DUN depend on specific applications. For SLP, the input includes mCSI and transmitted symbols of U users (i.e., $\mathcal{X} = \{\mathbf{h}_u, \xi_u \,|\, u \in \mathcal{U}\}$) and the label is the optimal transmit vector $\mathbf{x}^\star$ (i.e., $\mathcal{Y} = \{\mathbf{x}^\star\}$). The input and output for BLP can be similarly determined.

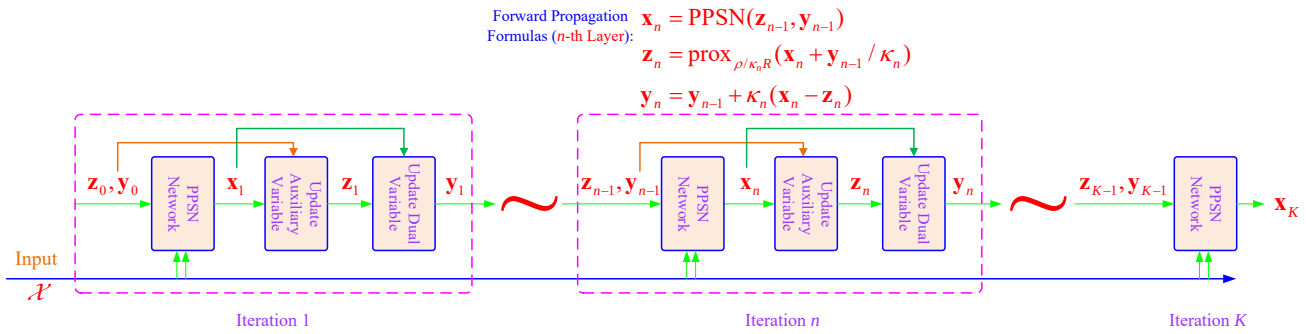

Fig. 3. The structure of the designed composite optimization algorithm induced deep-unfolding network (COA-DUN).

gradients required can be obtained via the chain rules manually or by the automatic differentiation methods supported by deep learning libraries (e.g., Pytorch and TensorFlow). The standard training procedure via mini-batch and first-order methods can be used to train the network. To improve sample efficiency and reduce sample complexity, it is recommended to choose the layer-wise or hierarchical training scheme or other pretraining methods. The method to train PPSN is similar.

Remark 4.1 In contrast to existing AU-based deep networks, which often require optimizing trainable matrix parameters, simple scalar parameters are optimized in this paper. Therefore, only a small number of training samples are required, which makes the proposed method eminently suitable for challenging rapidly fluctuating wireless environments. The reason for obtaining these benefits is that important prior knowledge (e.g., problem structures and algorithmic features) is captured and exploited when designing the networks.

V. Algorithm Unfolding for Symbol-Level Precoding

In this section, we employ the above techniques to tackle the SLP optimization problem. For convenience, the UPPiDO-based SLP optimization problem is rewritten as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \|\mathbf{F}\mathbf{x}\|^2 + \rho\|\mathbf{z}\|_1 \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{C}, \quad \mathbf{x} - \mathbf{z} = \mathbf{0}, \end{aligned} \tag{31}$$

where $\mathcal{C}$ denotes the CI-constraint based feasible set, i.e.,

$$\mathcal{C} = \Big\{\mathbf{x} \,\Big|\, \big|\mathrm{Im}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u})\big| \le \Big(\mathrm{Re}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u}) - \gamma_u\Big)\cdot \tan(\pi/K_u),\, \forall\, u \in \mathcal{U}\Big\}.$$

The augmented Lagrange penalty problem is given by

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \|\mathbf{F}\mathbf{x}\|^2 + \rho\|\mathbf{z}\|_1 + \frac{\kappa}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{y}/\kappa\|^2 \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{C}. \end{aligned} \tag{32}$$

Based on Algorithm 1, we can obtain an iterative algorithm immediately. In fact, the first sub-problem is given by

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{F}\mathbf{x}\|^2 + \frac{\kappa}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{y}/\kappa\|^2 \\ \text{s.t.} \quad & \big|\mathrm{Im}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u})\big| \le \Big(\mathrm{Re}(\mathbf{h}_u^{\mathrm{H}}\mathbf{x}e^{-j\xi_u}) - \gamma_u\Big)\cdot \\ & \tan(\pi/K_u),\ (\forall\, u \in \mathcal{U}). \end{aligned} \tag{33}$$

The problem in (33) is a quadratic program, which can be solved via a convex optimization toolbox. The second sub-problem (w.r.t. $\mathbf{z}$) is given by

$$\min_{\mathbf{z}} \quad \rho\|\mathbf{z}\|_1 + \frac{\kappa}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{y}/\kappa\|^2. \tag{34}$$

The closed-form solution of Problem (34) is given by

$$\mathbf{z}^\star = \mathrm{sgn}(\mathbf{x} + \mathbf{y}/\kappa) \odot \max\{|\mathbf{x} + \mathbf{y}/\kappa| - \rho/\kappa, 0\}. \tag{35}$$

An iterative algorithm, based on (33), (35) and Algorithm 1, is provided in Algorithm 3.

Algorithm 3: Iterative Algorithm for Problem (9)

1: **input:** estimated mCSI $\{\hat{\mathbf{h}}_u\}$, data symbols $\{\xi_u\}$, QoRS $\{\gamma_u\}$, codebook $\mathbf{F}$

2: **initialize** dual variable $\mathbf{y}$, auxiliary variable $\mathbf{z}$ and penalty parameter κ

3: **repeat**

(a) **update** primal variable $\mathbf{x}$ by solving (33)

(b) **update** auxiliary variable $\mathbf{z}$ as per (35)

(c) **update** dual variable $\mathbf{y}$ according to (20)

(d) **update** penalty parameter κ as per (21)

4: **until** some convergence criterion is met

5: **output:** optimal precoding vector $\mathbf{x}^\star$

Next, we proceed to unfold the iterative algorithm derived, whose task is to solve Problem (33). Because $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ are not holomorphic, the gradient of CI constraint function in (33) does not exist, and thus Algorithm 1 cannot be directly employed (in the complex-valued form). To tackle this issue, we explicitly convert Problem (33) into a real optimization problem. The conversion process is provided in Appendix A, and the problem obtained takes the form

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & \frac{1}{2}\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{P}\bar{\mathbf{x}} + \mathbf{q}_n^{\mathrm{T}}\bar{\mathbf{x}} \\ \text{s.t.} \quad & \mathbf{A}\bar{\mathbf{x}} \succeq \mathbf{b}, \end{aligned} \tag{36}$$

where $\bar{\mathbf{x}}$ collects the real and imaginary parts of variable $\mathbf{x}$. See Appendix A for details of Problem (36).

The Lagrange dual function of Problem (36) is given by

$$L(\bar{\mathbf{x}}, \boldsymbol{\mu}) = \frac{1}{2}\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{P}\bar{\mathbf{x}} + \mathbf{q}_n^{\mathrm{T}}\bar{\mathbf{x}} + \boldsymbol{\mu}^{\mathrm{T}}(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}). \tag{37}$$

According to (37), the primal-dual update formulas (within the n-th iteration) are simplified to:

$$\bar{\mathbf{x}}_m = \mathbf{P}^{-1}(\mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}_m - \mathbf{q}_n) \tag{38}$$

$$\boldsymbol{\mu}_{m+1} = \max\big\{\boldsymbol{\mu}_m + \alpha(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}_m), 0\big\}. \tag{39}$$

Let $\bar{\mathbf{x}}_{n,m}$ and $\boldsymbol{\mu}_{n,m+1}$ denote the optimization variable and dual variable of the m-th sub-iteration within the n-th iteration, respectively. Let furthermore $\mu_{n,m,i}$ represent the i-th component of $\boldsymbol{\mu}_{n,m}$. Then, the forward propagation formulas are given by

$$\bar{\mathbf{x}}_{n,m} = \mathbf{P}^{-1}(\mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}_{n,m} - \mathbf{q}_n) \tag{40}$$

$$\mu_{n,m+1,i} = \max\big\{\mu_{n,m,i} + \alpha_{n,m,i}(b_i - \mathbf{a}_i^{\mathrm{T}}\bar{\mathbf{x}}_{n,m}), 0\big\}, \tag{41}$$

where $\mathbf{a}_i^{\mathrm{T}}$ denotes the i-th row of matrix $\mathbf{A}$.

With (40) and (41) available and based on Algorithm 2, an efficient AU-aided deep network can be obtained immediately. Note that because the optimal solution of $\min_{\bar{\mathbf{x}}} L(\bar{\mathbf{x}}, \boldsymbol{\mu}_m)$ in this case has the closed-form expression, parameter B (within PPSN) is equal to 1 (i.e., $B = 1$), which thus simplifies the structure of COA-DUN. The trainable parameters are collected in $\boldsymbol{\Theta} = \{\alpha_{n,m,i}, \kappa_n \,|\, n = 1, \cdots, K, m = 1, \cdots, M, i = 1, \cdots, 2U\}$. The method of training $\boldsymbol{\Theta}$ has been provided in Algorithm 2, which is omitted to avoid repetition.

Remark 5.1 Since the CI-constraints in (33) are homogeneous (i.e., they are of the same type), if these constraints share the same parameters, i.e., $\alpha_{n,m,1} = \cdots = \alpha_{n,m,I}, (\forall\, m)$, the performance degeneration is negligible. However, the number of trainable parameters required is further reduced.

Finally, we analyze the computational complexity of the AU-based deep network, quantified by the number of multiplications. The computational complexity of the above AU-based deep network is on the order of $\mathcal{O}(8UG^2 + 8UGMK)$. If the regularization term $\rho R(\cdot)$ is absent, it reduces to $\mathcal{O}(8UG^2 + 8UGM)$. If the AU algorithm is directly applied to pCSI, the computational complexity further reduces to $\mathcal{O}(8UNM)$, which is scalable to both U (the number of served users) and N (the number of antennas). Since only simple scalar trainable parameters are learned and simple mathematical operations (e.g., matrix-vector multiplication and $\max(\cdot)$) are involved here, the resultant network has appealingly low computational complexity. These appealing features also make it possible to be implemented in real-time.

VI. Algorithm Unfolding for Block-Level Precoding

In this section, we employ the AU techniques developed in Section IV to address the sum-rate maximization based BLP problem, which is rewritten below for convenience

$$\begin{aligned}\max_{\{\mathbf{x}_u, \mathbf{z}_u\}} \ & \sum_{u\in\mathcal{U}} \log\left(1 + \frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\right|^2}{\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2 + \sigma^2}\right) - \rho\sum_{u\in\mathcal{U}}\|\mathbf{z}_u\|_1 \\ \text{s.t.} \quad & \mathbf{x}_u = \mathbf{z}_u, \ (\forall\, u \in \mathcal{U}), \quad \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2 \le P_{\mathrm{b}}.\end{aligned} \tag{42}$$

The augmented Lagrange penalty problem is given by

$$\begin{aligned}\max_{\{\mathbf{x}_u, \mathbf{z}_u\}} \ & \sum_{u\in\mathcal{U}} \log\left(1 + \frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\right|^2}{\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2 + \sigma^2}\right) \\ & \qquad - \rho\sum_{u\in\mathcal{U}}\|\mathbf{z}_u\|_1 - \frac{\kappa}{2}\sum_{u\in\mathcal{U}}\left\|\mathbf{x}_u - \mathbf{z}_u + \frac{\mathbf{y}_u}{\kappa}\right\|^2 \\ \text{s.t.} \quad & \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2 \le P_{\mathrm{b}}.\end{aligned} \tag{43}$$

To solve Problem (43), we introduce $2U$ auxiliary variables $\{p_u, q_u\}$ and equivalently write Problem (43) as

$$\begin{aligned}\max_{\mathcal{V}} \ & \sum_{u\in\mathcal{U}}\left(\log(1+p_u) - \rho\|\mathbf{z}_u\|_1 - \frac{\kappa}{2}\left\|\mathbf{x}_u - \mathbf{z}_u + \frac{\mathbf{y}_u}{\kappa}\right\|^2\right) \\ \text{s.t.} \ & \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2 \le P_{\mathrm{b}}, \quad \frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\right|^2}{q_u} \ge p_u \\ & \sigma^2 + \sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2 \le q_u, \ (\forall\, u \in \mathcal{U}),\end{aligned} \tag{44}$$

where $\mathcal{V} = \{\mathbf{x}_u, \mathbf{z}_u, p_u, q_u\}$ is introduced for simplicity.

Let $\mathbf{x}_{u,n}$ denote the n-th iteration of $\mathbf{x}_u$, and the notations are defined similarly for other variables. Then, $\mathbf{x}_{u,n+1}$ can be obtained by solving the convex optimization problem in (46) (the top of next page). Note that the form in (46) has satisfied the requirements of applying Algorithm 1 and can be solved efficiently. Specifically, the first sub-problem is given by

$$\begin{aligned}\max_{\{\mathbf{x}_u, p_u, q_u\}} \ & \sum_{u\in\mathcal{U}}\left(\log(1+p_u) - \frac{\kappa_n}{2}\left\|\mathbf{x}_u - \mathbf{z}_u + \frac{\mathbf{y}_u}{\kappa_n}\right\|^2\right) \\ \text{s.t.} \ & \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2 \le P_{\mathrm{b}} \\ & \frac{2\mathrm{Re}\big(\mathbf{x}_{u,n}^{\mathrm{H}}\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\big)}{q_{u,n}} - \frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n}\right|^2 q_u}{q_{u,n}^2} \ge p_u \\ & \sigma^2 + \sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2 \le q_u, \ (\forall\, u \in \mathcal{U}).\end{aligned} \tag{45}$$

The problem in (45) is convex and can be efficiently solved.

The second sub-problem is given by

$$\min_{\{\mathbf{z}_u\}} \ \sum_{u\in\mathcal{U}}\left(\rho\|\mathbf{z}_u\|_1 + \frac{\kappa_n}{2}\left\|\mathbf{x}_u - \mathbf{z}_u + \frac{\mathbf{y}_u}{\kappa_n}\right\|^2\right). \tag{50}$$

Note that the optimization problem in (50) is separable, and the optimal solution is given by

$$\mathbf{z}_u^{\star} = \mathrm{sgn}(\mathbf{x}_u + \mathbf{y}_u/\kappa_n) \odot \max\{|\mathbf{x}_u + \mathbf{y}_u/\kappa_n| - \rho/\kappa_n, 0\}. \tag{51}$$

According to (20), each dual variable $\mathbf{y}_u$ is updated as

$$\mathbf{y}_{u,n+1} = \mathbf{y}_{u,n} + \kappa_n(\mathbf{x}_{u,n} - \mathbf{z}_{u,n}), \tag{52}$$

while the penalty parameter κ is updated as

$$\kappa_{n+1} = \begin{cases} \beta\kappa_n & \sum_{u\in\mathcal{U}}\|\mathbf{x}_{u,n} - \mathbf{z}_{u,n}\|^2 > \\ & \qquad \gamma\sum_{u\in\mathcal{U}}\|\mathbf{x}_{u,n-1} - \mathbf{z}_{u,n-1}\|^2 \\ \kappa_n & \text{otherwise.}\end{cases} \tag{53}$$

$$\begin{aligned}\max_{\mathcal{V}} \quad & \sum_{u\in\mathcal{U}}\bigg(\log\left(1+p_u\right)-\rho\|\mathbf{z}_u\|_1-\frac{\kappa_n}{2}\left\|\mathbf{x}_u-\mathbf{z}_u+\frac{\mathbf{y}_u}{\kappa_n}\right\|^2\bigg)\\ \text{s.t.} \quad & \sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2\le P_{\mathrm{b}},\ \ \frac{2\mathrm{Re}\big(\mathbf{x}_{u,n}^{\mathrm{H}}\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\big)}{q_{u,n}}-\frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n}\right|^2 q_u}{q_{u,n}^2}\ge p_u,(\forall\, u\in\mathcal{U}),\ \ \sigma^2+\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2\le q_u,(\forall\, u\in\mathcal{U}).\end{aligned} \tag{46}$$

$$\begin{aligned}L(\{\mathbf{x}_u,p_u,q_u\})=&\sum_{u\in\mathcal{U}}\bigg(-\log\left(1+p_u\right)+\frac{\kappa_n}{2}\left\|\mathbf{x}_u-\mathbf{z}_u+\frac{\mathbf{y}_u}{\kappa_n}\right\|^2\bigg)+\lambda\bigg(-P_{\mathrm{b}}+\sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_u\|^2\bigg)\\&+\sum_{u\in\mathcal{U}}\nu_u\bigg(\sigma^2-q_u+\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_v\right|^2\bigg)+\sum_{u\in\mathcal{U}}\tau_u\bigg(p_u-\frac{2\mathrm{Re}\big(\mathbf{x}_{u,n}^{\mathrm{H}}\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_u\big)}{q_{u,n}}+\frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n}\right|^2 q_u}{q_{u,n}^2}\bigg).\end{aligned} \tag{47}$$

$$\begin{aligned}\frac{\partial L}{\partial\mathbf{x}_u^{\mathrm{H}}}&=\bigg(\frac{\kappa_n}{2}\mathbf{I}+\lambda\mathbf{F}^{\mathrm{H}}\mathbf{F}+\sum_{v\neq u}\nu_v\mathbf{h}_v\mathbf{h}_v^{\mathrm{H}}\bigg)\mathbf{x}_u-\frac{\tau_u\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n}}{q_{u,n}}+\frac{\kappa_n}{2}\bigg(\frac{\mathbf{y}_u}{\kappa_n}-\mathbf{z}_u\bigg),\\ \frac{\partial L}{\partial p_u}&=\tau_u-\frac{1}{1+p_u},\ \ \frac{\partial L}{\partial q_u}=-\nu_u+\frac{\tau_u\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n}\right|^2}{q_{u,n}^2}.\end{aligned} \tag{48}$$

$$\begin{aligned}\mathbf{x}_{u,n,m,l}=&\mathbf{x}_{u,n,m,l-1}-a_{u,n,m,l}\frac{\partial L}{\partial\mathbf{x}_u^{\mathrm{H}}}\bigg|_{\mathbf{x}_u=\mathbf{x}_{u,n,m,l-1}},\quad p_{u,n,m}=p_{u,n,m-1}-b_{u,n,m}\frac{\partial L}{\partial p_u}\bigg|_{p_u=p_{u,n,m-1}}\\ q_{u,n,m}=&q_{u,n,m-1}-c_{u,n,m}\frac{\partial L}{\partial q_u},\quad \lambda_{n,m+1}=\max\bigg\{\lambda_{n,m}+\alpha_{n,m}\bigg(-P_{\mathrm{b}}+\sum_{u\in\mathcal{U}}\|\mathbf{F}\mathbf{x}_{u,n,m,B}\|^2\bigg),0\bigg\}\\ \nu_{u,n,m+1}=&\max\bigg\{\nu_{u,n,m}+\zeta_{u,n,m}\bigg(\sigma^2-q_{u,n,m}+\sum_{v\neq u}\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{v,n,m,B}\right|^2\bigg),0\bigg\}\\ \tau_{u,n,m+1}=&\max\bigg\{\tau_{u,n,m}+\xi_{u,n,m}\bigg(p_{u,n,m}-\frac{2\mathrm{Re}\big(\mathbf{x}_{u,n-1,M,B}^{\mathrm{H}}\mathbf{h}_u\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n,m,B}\big)}{q_{u,n-1,M}}+\frac{\left|\mathbf{h}_u^{\mathrm{H}}\mathbf{x}_{u,n-1,M,B}\right|^2 q_{u,n,m}}{q_{u,n-1,M}^2}\bigg),0\bigg\}.\end{aligned} \tag{49}$$

Algorithm 4: Iterative Algorithm for Problem (10)

1: **input:** estimated mCSI $\{\hat{\mathbf{h}}_u\}$, codebook $\mathbf{F}$, maximal transmit power P_{b}

2: **initialize** dual variables $\{\mathbf{y}_u\}$, auxiliary variables $\{\mathbf{z}_u\}$ and penalty parameter κ

3: **repeat**

(a) **update** variables $\{\mathbf{x}_u,p_u,q_u\}$ by solving (45)

(b) **update** auxiliary variables $\{\mathbf{z}_u\}$ as per (51)

(c) **update** dual variables $\{\mathbf{y}_u\}$ according to (52)

(d) **update** penalty parameter κ according to (53)

4: **until** some convergence criterion is met

5: **output:** optimal precoding vectors $\{\mathbf{x}_u^\star\}$

For clarity, the complete iterative procedure solving Problem (10) is provided in Algorithm 4.

Now, we proceed to unfold the iterative algorithm obtained. It may be readily shown that the key is to unfold the iterative algorithm that solves Problem (45). The Lagrangian of Problem (45), denoted by $L(\{\mathbf{x}_u,p_u,q_u\})$, is provided in (47), where λ, ν_u and τ_u represent the dual variables for the corresponding inequality constraints. Given λ, $\{\nu_u\}$ and $\{\tau_u\}$, the partial derivatives of $L(\{\mathbf{x}_u,p_u,q_u\})$ (w.r.t. $\mathbf{x}_u^{\mathrm{H}}$, p_u and q_u) are given in (48). With the partial derivatives (or gradients) available, we can next design or construct the PPSN.

Let $\mathbf{x}_{u,n,m,l}$ represent the optimization variable of user u of the l-th sub-iteration of the m-th PPSN subnetwork of the n-th outer layer or iteration, and the other notations are defined similarly. Given (48), the forward propagation formulas (in (49)) and thus the PPSN can be immediately obtained. The trainable parameters of M PPSN subnetworks within the n-th layer or iteration are collected in set $\mathbf{\Theta}_n$ $(1\le n\le K)$, i.e.,

$$\begin{aligned}\mathbf{\Theta}_n=\{&a_{u,n,m,l},b_{u,n,m},c_{u,n,m},\alpha_{n,m},\xi_{u,n,m},\zeta_{u,n,m}\,|\\&1\le u\le U,1\le m\le M,1\le l\le B\}.\end{aligned}$$

In general, different primal or dual updates should use different (trainable) step-size parameters, so as to achieve better convergence performance. But, if necessary, these primal or dual updates can share a part of trainable step-size parameters (e.g., $\{b_{u,n,m}=c_{u,n,m}\}$ and $\{\xi_{u,n,m}=\zeta_{u,n,m}\}$), which can significantly reduce the scale of trainable parameters. The trainable parameters are collected in $\mathbf{\Theta}$, i.e.,

$$\mathbf{\Theta}=\{\mathbf{\Theta}_1,\mathbf{\Theta}_2,\cdots,\mathbf{\Theta}_K,\kappa_1,\kappa_2,\cdots,\kappa_{K-1}\}. \tag{54}$$

Now, an efficient COA-DUN has been obtained, which can be trained via various training procedures.

Finally, we analyze the computational complexity of the COA-DUN obtained, still quantified by the number of multiplications. The computational complexity of each PPSN (e.g., when the regularization term $\rho R(\cdot)$ is absent) is dominated by $\mathcal{O}\big[(4NG^2+8G^2+8G)BMU+(4NG^2+8G)MU\big]$. Then, the overall computational complexity of COA-DUN is $\mathcal{O}\big[(4NG^2+8G^2+8G)BMUK+(4NG^2+8G)MUK\big]$. To simplify the system design, B can often be set to 1 (i.e., $B=1$). In this case, the computational complexity reduces to $\mathcal{O}\big((8NG^2+8G^2+16G)MUK\big)$.

VII. Simulation Results

In this section, simulation results are provided for characterizing the performance and verifying the superiority of the proposed algorithms. The codebooks are obtained by uniformly sampling the beam space $[-1,1]$ [46]. Specifically, a codebook of size G is constructed as follows:

$$\mathcal{F}=\big\{\mathbf{a}_i\,\big|\,\mathbf{a}_i=\mathbf{a}\big(-1+2i/G\big),i=0,\cdots,G-1\big\}.$$

To ensure the positivity of the trainable step-size parameters, the softplus function, i.e., $\ln[1+\exp(\cdot)]$, is chosen to reparameterize the learnable parameters introduced. During the process of estimating mCSI or pCSI and feeding it back to the BS, it is inevitable to incur estimation errors. The estimated mCSI $\hat{\mathbf{h}}_u$ and accurate mCSI $\mathbf{h}_u$ satisfy

$$\mathbf{h}_u=\mathbf{F}^{\mathrm{H}}\bar{\mathbf{h}}_u=\hat{\mathbf{h}}_u+\Delta\mathbf{h}_u, \tag{55}$$

where $\Delta\mathbf{h}_u$ is distributed as $\Delta\mathbf{h}_u\sim\mathcal{CN}(0,\sigma_h^2\mathbf{I})$ [14]. It takes a similar form for pCSI.

The number of users communicating to the BS is assumed to $U=3$. For the channel model in (1), the power angular spectrum function with the uniform distribution [41] is chosen to evaluate different precoding algorithms. To evaluate the generalization performance of our methods, we consider three cases of angular spreads for the U users:

Case 1: $[-33/64,-27/64],[-3/64,3/64],[27/64,33/64]$

Case 2: $[-20/64,-14/64],[-3/64,3/64],[14/64,20/64]$

Case 3: $[-24/64,-13/64],[-6/64,6/64],[13/64,24/64]$.

Unless otherwise specified, Case 1 is chosen to train the AU-based precoding models, while the other two cases are used for evaluating different precoding algorithms.

The benchmarks used to evaluate our algorithms, as well as their basic characteristics are provided in Table II. For convenience, the SLP algorithm based on the composite optimization algorithm (i.e., Algorithm 1) is termed as COA-SLP. For BLP, the WMMSE algorithm and its AU-based version in [26] are chosen as benchmarks. The precoding algorithm developed in this paper (based on UPPiDO and successive convex approximation) is referred to as COA-SCAP. The BLP algorithm without UPPiDO is referred to as SCAP. When the AU techniques are applied to an algorithm, denoted by ALGO, the AU-based algorithm obtained is referred to as ALGO-AU. Unless otherwise specified, an abbreviation without "AU" indicates that the algorithm is derived via IPM.

The normalized mean square error (NMSE), symbol error rate (SER), average achievable sum-rate (AASR) and running time are chosen as performance metrics to evaluate the different algorithms. The NMSE is defined as

$$\mathrm{NMSE}=10\log\frac{\|\hat{\mathbf{x}}-\mathbf{x}^\star\|^2}{\|\mathbf{x}^\star\|^2},$$

where $\hat{\mathbf{x}}$ and $\mathbf{x}^\star$ denote the predicted precoder and optimal precoder, respectively. The NMSE characterizes the quality of precoder recovered by a precoding algorithm, which directly and intuitively characterizes its performance. As for SER and AASR, they characterize the communication performance.

$\mathcal{D}$ denotes the training dataset, and D is the size of this set. The SNR for BLP is defined by P_{b}/σ^2, with σ^2 normalized to 1. For SLP, the standard training procedure is utilized to train the AU-based precoding network, i.e., all parameters are updated simultaneously. For BLP, the layer-wise training method is chosen to train the precoding network, e.g., the AU network is optimized layer by layer with the same dataset. [5]

A. Performance Evaluation for Symbol-Level Precoding

In this subsection, we evaluate the performance of different CI-based SLP algorithms. An appealing advantage of AU is that it has low computational complexity. To show this, we evaluate the NMSE performance of FS+SLP and FS+SLP-AU, as shown in Fig. 4. For fairness, here the primal-dual method is used for solving the precoding problem from FS+SLP, and "FS+SLP" is replaced by "FS+SLP-PDM" to avoid confusion. It is observed that FS+SLP-AU outperforms FS+SLP-PDM. Specifically, to achieve the same NMSE performance, FS+SLP-AU requires about one third of the iterations. The reason for this benefit is that FS+SLP-AU can adapt to the data (and further the environment). It is also observed that although the number of layers is relatively small (e.g., $M=6$), good NMSE performance can still be achieved.

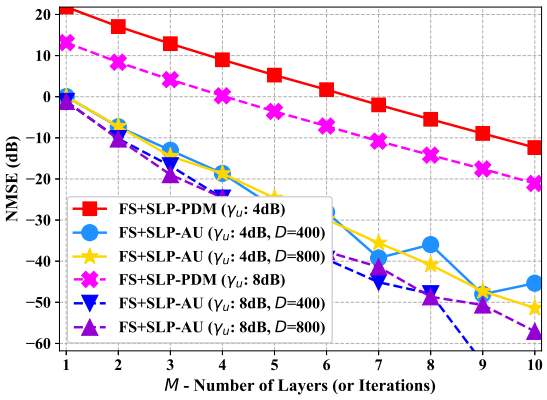

Fig. 4. The average test NMSE performance of FS+SLP-AU and FS+SLP-PDM (i.e., the primal-dual method is chosen to solve the precoding optimization problem): $N=G=64$, QPSK modulation and Case 1.

Fig. 5 shows the SER performance of different SLP algorithms. It is observed that COA-SLP outperforms both FD-SLP

[5]To promote reproducible research, the codes for generating the simulation results in the paper (as well as the convergence analysis of our algorithms) are made available on the github website, i.e., the TWC2023-DU branch.

Type	Abbreviation	Full Name and Basic Characteristic	Unfolded Solution ?	CSI Required
SLP	FD-SLP	(Classical) Fully-Digital SLP [14]	No	pCSI $\{\mathbf{h}_u\}$
	FS+SLP	Fully-Sweeping [46] + SLP	No	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	COA-SLP	Composite Optimization (or UPPiDO) + SLP	No	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	FS+SLP-AU	Fully-Sweeping + SLP + AU	Yes	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	COA-SLP-AU	UPPiDO + SLP + AU	Yes	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
BLP	WMMSE	Weighted MMSE (Precoding)	No	pCSI $\{\mathbf{h}_u\}$
	SCAP	Successive Convex Approximation + BLP	No	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	COA-SCAP	UPPiDO + SCA + BLP	No	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	WMMSE-AU	Weighted MMSE + AU [26]	Yes	pCSI $\{\mathbf{h}_u\}$
	SCAP-AU	SCA + BLP + AU	Yes	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$
	COA-SCAP-AU	UPPiDO + SCA + BLP + AU	Yes	mCSI $\{\mathbf{F}^{\mathrm{H}}\mathbf{h}_u\}$

TABLE II
BASIC INFORMATION OF TYPICAL PRECODING ALGORITHMS FOR PERFORMANCE EVALUATION

and FS+SLP, which verifies the effectiveness of the UPPiDO-based precoding framework (i.e., it helps to achieve more robust performance for inaccurate mCSI). The reason for this is two-fold. On the one hand, significant mCSI that is key to designing precoders is identified and estimated. On the other hand, insignificant inaccurate mCSI that may degrade the precoder is discarded, which alleviates the noise amplification effect. In addition to the good SER performance, another important advantage of COA-SLP is that the overhead in terms of training and feedback is much lower than that of FD-SLP or FS+SLP [18]. For the basic principle of this advantage, we refer interested readers to [18] for the details.

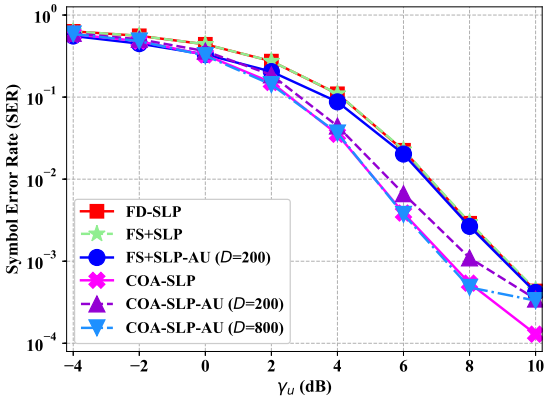

Fig. 5. The SER performance of different SLP precoding algorithms: $N = G = 64$, $\sigma_h = 0.4$, $\rho = 5$, QPSK modulation and Case 1.

It is also observed from Fig. 5 that with a relatively small number of training samples (e.g., $D = 800$ for COA-SLP-AU and $D = 200$ for FS+SLP-AU), the AU-based algorithms can still achieve almost the same SER performance as their optimization-based counterparts. For COA-SLP-AU, good SER performance (and better than FS+SLP) can still be achieved even when the size of the training dataset is as small as $D = 200$. The reason for this is that the deep network models obtained via the AU method in this paper maintain and exploit prior information of the underlying optimization problems and iterative algorithms (e.g., problem structures and algorithmic features) as much as possible. As a result, the trainable parameters are scalars (instead of matrices), which alleviates the requirement of training samples. Note that in practice it may be expensive to collect training samples, and thus the proposed AU approach is appealing.

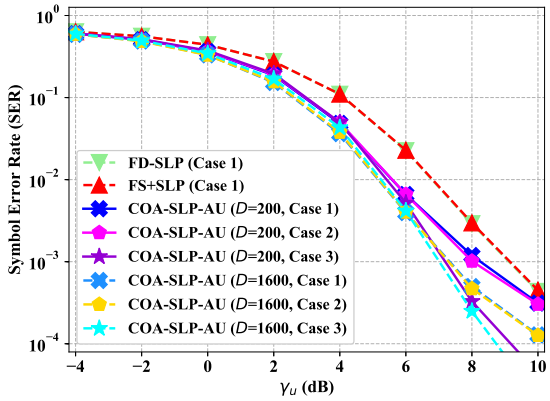

Fig. 6. The generalization performance of the designed AU algorithm under different channel environments: $N = G = 64$, $\sigma_h = 0.4$, $\rho = 5$ and QPSK.

In contrast to other machine learning applications, the channel environments of practical wireless communication systems often change very fast, which increases the difficulties of both collecting training samples and training learning models. Therefore, it is hoped that the learning models have a good generalization behavior. Fortunately, the learning models constructed via the proposed AU method naturally have good generalization performance, as shown in Fig. 6. It is observed that although the learning models (i.e., the unfolded deep networks) are trained based on the first kind of channel environments (i.e., Case 1), they can achieve good SER performance in the other two channel environments. The reason for this is that the trainable parameters of the AU models obtained via our AU approach are scalar step-size parameters, which mainly affect the convergence rate of the iterative algorithm.

B. Performance Evaluation for Block-Level Precoding

In the second subsection, we evaluate the performance of different (sum-rate maximization based) BLP algorithms. It is widely recognized that the size of training dataset profoundly

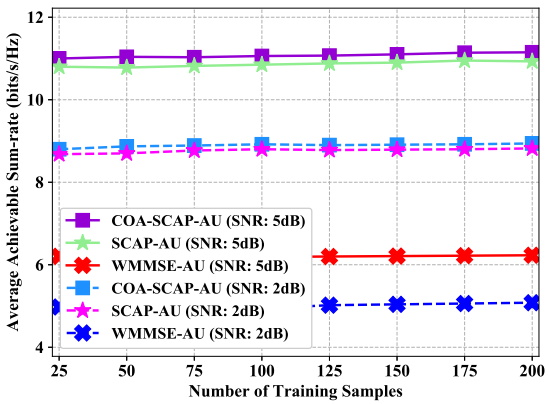

Fig. 7. The AASR performance vs the number of training samples used to train AU models: $N = G = 64$, $\sigma_h = 0.4$, $\rho = 0.1$ and Case 1.

affects the performance of an AI algorithm. Fortunately, an important advantage of our AU approach is that good small-sample performance can be achieved, as shown in Fig. 7. The reason for this is two-fold. On the one hand, important prior or domain specific knowledge of the iterative algorithm has been incorporated when designing the algorithmic network. On the other hand, the trainable parameters are all scalars, which further reduces the number of samples required. It is observed that COA-SCAP-AU and SCAP-AU outperform WMMSE-AU, since the trainable parameters of WMMSE-AU contain many matrices and thus it requires more samples.

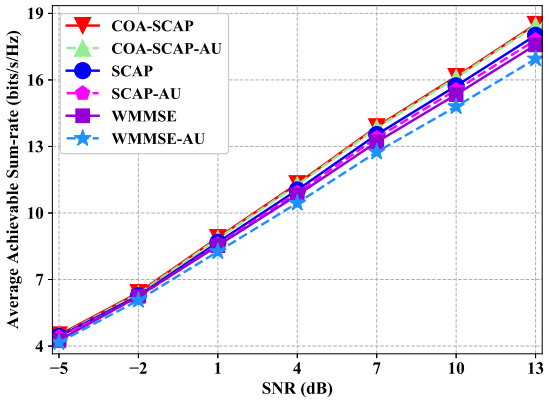

Fig. 8. The AASR performance of different BLP precoding algorithms: $N = G = 64$, $\sigma_h = 0.4$, $\rho = 0.1$ and Case 1.

The AASR performance of different precoding algorithms is shown in Fig. 8. Similar to the SLP case, it is observed that the UPPiDO-based algorithms (including COA-SCAP and COA-SCAP-AU) outperform the other precoding algorithms, which further verifies the superiority of the UPPiDO-based precoding framework. In addition to the good AASR performance and low training as well as feedback overhead, the low computational complexity of COA-SCAP-AU can be obtained with a relatively small number of training samples. The key reason is that important prior information/knowledge has been retained when unfolding the iterative algorithms and the trainable parameters are all scalar step-size parameters. The benefit is that the size of the training dataset can be safely reduced, which also reduces the training complexity.

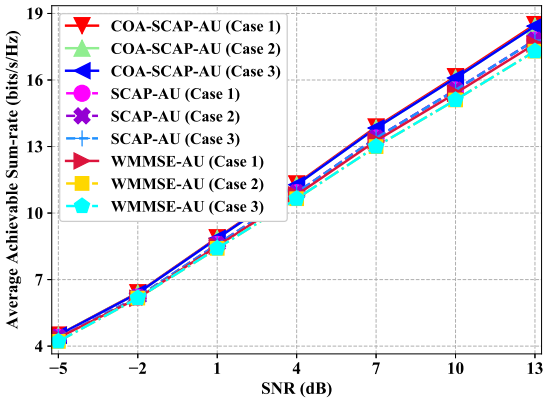

Fig. 9. The generalization performance of the designed AU algorithm under different channel environments: $N = G = 64$ and $\sigma_h = 0.4$.

As we have mentioned before, another important advantage of the proposed AU method is that the AU-based deep network conceived has good natural generalization performance. Fig. 9 demonstrates the generalization performance of the proposed AU-based algorithm under different channel environments. It is seen that the AU-based deep networks trained on a specific channel environment perform well on the other two. It is also seen that COA-SCAP-AU outperforms WMMSE-AU.

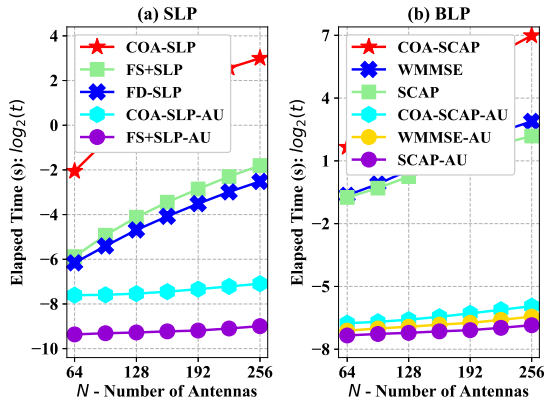

Fig. 10. The average elapsed time of different precoding algorithms (solving one problem): $N = G$, $\sigma_h = 0.4$ and Case 1.

To intuitively demonstrate the advantage of the AU techniques in terms of low computational complexity, the average run-time of different precoding algorithms (to solve one problem) for the two precoding settings is shown in Fig. 10, which is obtained by averaging over 5×10^5 (for SLP) and 5×10^3 (for BLP) channel realizations, respectively. It is seen that the AU-based precoding algorithms (e.g., COA-SLP-AU and COA-SCAP-AU) run much faster than their iteration-based counterparts. More appealingly, as the number of transmit antennas increases, this advantage becomes more pronounced. The reason for this is that, as an example, the AU-based precoding algorithm COA-SLP-AU has a complexity order $\mathcal{O}(8UG^2 + 8UGMK)$, rather than in excess of $\mathcal{O}(G^3)$ as required for the iteration-based counterpart COA-SLP.

VIII. Conclusion

To reduce the computational complexity of the UPPiDO-based designs proposed recently, in this paper we conceived an efficient approach to unfold a tailored iterative optimization algorithm for the family of UPPiDO-based optimization problems. First, we proposed an unfolding-friendly iterative optimization algorithm for a family of non-convex and non-smooth problems extensively encountered both in signal processing and machine learning. Then, we unfolded the iterative optimization algorithm by conceiving an efficient iterative algorithm for solving the sub-problems and treating important hyper-parameters as learnable parameters to obtain a deep network. As an example, we applied the proposed AU techniques to the SLP and BLP optimization problems. Both complexity analysis and simulation results were provided to confirm the effectiveness and superiority of our proposals.

Appendix A
Conversion of SLP with CI Constraints into Quadratic Program

To convert Problem (33) into a real-valued problem, we define $\mathbf{x} = \mathbf{x}_{\mathrm{R}} + j\mathbf{x}_{\mathrm{I}}$ and $e^{j\xi_u}\mathbf{h}_u = \tilde{\mathbf{h}}_{\mathrm{R},u} + j\tilde{\mathbf{h}}_{\mathrm{I},u}$, where $\mathbf{x}_{\mathrm{R}}$ (or $\mathbf{x}_{\mathrm{I}}$) represents the real (or imaginary) part of the optimization variable $\mathbf{x}$ and the real vectors $\tilde{\mathbf{h}}_{\mathrm{R},u}$ and $\tilde{\mathbf{h}}_{\mathrm{I},u}$ are defined similarly. Furthermore, by letting $c_u = \tan(\pi/K_u)$, $\mathbf{Q}_{\mathrm{R}} = 2\mathrm{Re}(\mathbf{F}^{\mathrm{H}}\mathbf{F}) + \kappa_n\mathbf{I}$, $\mathbf{Q}_{\mathrm{I}} = 2\mathrm{Im}(\mathbf{F}^{\mathrm{H}}\mathbf{F})$, $\mathbf{q}_n^{\mathrm{R}} = \mathrm{Re}(\mathbf{y}_{n-1} - \kappa_n\mathbf{z}_{n-1})$ and $\mathbf{q}_n^{\mathrm{I}} = \mathrm{Im}(\mathbf{y}_{n-1} - \kappa_n\mathbf{z}_{n-1})$, Problem (33), which is used for generating the n-th iteration of $\mathbf{x}$ can be equivalently written as

$$\begin{aligned}\min_{\mathbf{x}_{\mathrm{R}},\mathbf{x}_{\mathrm{I}}}\ & \frac{1}{2}\begin{bmatrix}\mathbf{x}_{\mathrm{R}}\\ \mathbf{x}_{\mathrm{I}}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{Q}_{\mathrm{R}} & -\mathbf{Q}_{\mathrm{I}}\\ \mathbf{Q}_{\mathrm{I}} & \mathbf{Q}_{\mathrm{R}}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\mathrm{R}}\\ \mathbf{x}_{\mathrm{I}}\end{bmatrix} + \begin{bmatrix}\mathbf{q}_n^{\mathrm{R}}\\ \mathbf{q}_n^{\mathrm{I}}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{x}_{\mathrm{R}}\\ \mathbf{x}_{\mathrm{I}}\end{bmatrix}\\ \text{s.t.}\ & \begin{bmatrix}c_u\tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}} + \tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}}, & c_u\tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}} - \tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}}\\ c_u\tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}} - \tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}}, & c_u\tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}} + \tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{\mathrm{R}}\\ \mathbf{x}_{\mathrm{I}}\end{bmatrix} \succeq \\ & \qquad \begin{bmatrix}c_u\gamma_u\\ c_u\gamma_u\end{bmatrix},\ (\forall\, u \in \mathcal{U}).\end{aligned} \tag{56}$$

To facilitate our further discussion, we write Problem (56) into a more compact form. Specifically, we define matrices $\mathbf{P}$, $\mathbf{A}$, vectors $\mathbf{q}_n$, $\mathbf{b}$ and variable $\bar{\mathbf{x}}$ as follows:

$$\begin{aligned}&\mathbf{b} = [c_1\gamma_1, c_1\gamma_1, c_2\gamma_2, c_2\gamma_2, \cdots, c_U\gamma_U, c_U\gamma_U]^{\mathrm{T}} \in \mathbb{R}^{2U},\\ &\mathbf{P} = \begin{bmatrix}\mathbf{Q}_{\mathrm{R}} & -\mathbf{Q}_{\mathrm{I}}\\ \mathbf{Q}_{\mathrm{I}} & \mathbf{Q}_{\mathrm{R}}\end{bmatrix} \in \mathbb{R}^{2G\times 2G},\ \mathbf{q}_n = \begin{bmatrix}\mathbf{q}_n^{\mathrm{R}}\\ \mathbf{q}_n^{\mathrm{I}}\end{bmatrix},\ \bar{\mathbf{x}} = \begin{bmatrix}\mathbf{x}_{\mathrm{R}}\\ \mathbf{x}_{\mathrm{I}}\end{bmatrix},\\ &\mathbf{A} = \begin{bmatrix}\mathbf{A}_1\\ \vdots\\ \mathbf{A}_U\end{bmatrix} \text{ with } \mathbf{A}_u = \begin{bmatrix}c_u\tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}} + \tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}}, & c_u\tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}} - \tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}}\\ c_u\tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}} - \tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}}, & c_u\tilde{\mathbf{h}}_{\mathrm{I},u}^{\mathrm{T}} + \tilde{\mathbf{h}}_{\mathrm{R},u}^{\mathrm{T}}\end{bmatrix}.\end{aligned}$$

Given the above notations, Problem (56) can be rewritten as

$$\begin{aligned}\min_{\bar{\mathbf{x}}}\ & \frac{1}{2}\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{P}\bar{\mathbf{x}} + \mathbf{q}_n^{\mathrm{T}}\bar{\mathbf{x}}\\ \text{s.t.}\ & \mathbf{A}\bar{\mathbf{x}} \succeq \mathbf{b}.\end{aligned} \tag{57}$$

References

[1] 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.

[2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, 2013.

[3] Y. Wang, X. Chen, Y. Cai, and L. Hanzo, "RIS-aided hybrid massive MIMO systems relying on adaptive-resolution ADCs: Robust beamforming design and resource allocation," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3281–3286, 2022.

[4] A. A. Nasir, H. D. Tuan, T. Q. Duong, H. V. Poor, and L. Hanzo, "Hybrid beamforming for multi-user millimeter-wave networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2943–2956, 2020.

[5] W. Huang, Y. Zeng, and Y. Huang, "Achievable rate region of MISO interference channel aided by intelligent reflecting surface," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16 264–16 269, 2020.

[6] X. Yu, J. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 485–500, 2016.

[7] J. Zhang, Y. Huang, J. Wang, and L. Yang, "Hybrid precoding for wideband millimeter-wave systems with finite resolution phase shifters," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 285–11 290, Nov 2018.

[8] C. Masouros and E. Alsusa, "Dynamic linear precoding for the exploitation of known interference in MIMO broadcast systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1396–1404, 2009.

[9] C. Masouros, "Correlation rotation linear precoding for MIMO broadcast communications," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 252–262, 2011.

[10] C. Masouros and T. Ratnarajah, "Interference as a source of green signal power in cognitive relay assisted co-existing MIMO wireless transmissions," *IEEE Trans. Commun.*, vol. 60, no. 2, pp. 525–536, 2012.

[11] C. Masouros, T. Ratnarajah, M. Sellathurai, C. B. Papadias, and A. K. Shukla, "Known interference in the cellular downlink: a performance limiting factor or a source of green signal power?" *IEEE Commun. Mag.*, vol. 51, no. 10, pp. 162–171, 2013.

[12] C. Masouros, M. Sellathurai, and T. Ratnarajah, "Vector perturbation based on symbol scaling for limited feedback MISO downlinks," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 562–571, 2014.

[13] M. Alodeh, S. Chatzinotas, and B. Ottersten, "Constructive multiuser interference in symbol level precoding for the MISO downlink channel," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2239–2252, 2015.

[14] C. Masouros and G. Zheng, "Exploiting known interference as green signal power for downlink beamforming optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3628–3640, 2015.

[15] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, 2016.

[16] X. Gao, L. Dai, S. Han, C. I, and R. W. Heath, "Energy-efficient hybrid analog and digital precoding for MmWave MIMO systems with large antenna arrays," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 998–1009, 2016.

[17] S. He, Y. Huang, J. Wang, L. Yang, and W. Hong, "Joint antenna selection and energy-efficient beamforming design," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1165–1169, 2016.

[18] J. Zhang and C. Masouros, "A unified framework for precoding and pilot design for FDD symbol-level precoding," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2862–2875, 2022.

[19] M. Hong, R. Sun, H. Baligh, and Z.-Q. Luo, "Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 226–240, 2013.

[20] W. Huang, Y. Huang, R. Zhao, S. He, and L. Yang, "Wideband millimeter wave communication: Single carrier based hybrid precoding with sparse optimization," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9696–9710, 2018.

[21] 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.

[22] 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.

[23] 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, 2020.

[24] W. Lee, M. Kim, and D. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, 2018.

[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[26] 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, 2021.

[27] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Viterbinet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, 2020.

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

[29] M. Borgerding, P. Schniter, and S. Rangan, "AMP-Inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, 2017.

[30] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, 2019.

[31] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-driven symbol detection via model-based machine learning," *Communications in Information and Systems*, vol. 20, no. 3, pp. 283–317, 2020.

[32] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-based deep learning: On the intersection of deep learning and optimization," *IEEE Access*, vol. 10, pp. 115 384–115 398, 2022.

[33] A. Agrawal, S. Barratt, and S. Boyd, "Learning convex optimization models," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 8, pp. 1355–1364, 2021.

[34] A. Agrawal, S. T. Barratt, S. P. Boyd, and B. Stellato, "Learning convex optimization control policies," in *Proceedings of Machine Learning Research*, 2020, pp. 361–373.

[35] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Trans. Signal Process.*, vol. 70, pp. 1532–1547, 2022.

[36] A. Mohammad, C. Masouros, and Y. Andreopoulos, "An unsupervised learning-based approach for symbol-level-precoding," in *2021 IEEE GLOBECOM*, 2021, pp. 1–6.

[37] J. Zhang and C. Masouros, "A deep-learning based framework for joint downlink precoding and CSI sparsification," in *2022 IEEE International Conference on Communications (ICC)*, 2022, pp. 1–6.

[38] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, vol. 36, 09 2019.

[39] J. Gallier and J. Quaintance, *Fundamentals of Optimization Theory with Applications to Machine Learning*. Linear Algebra and Optimization with Applications to Machine Learning, 2020.

[40] A. Ruszczynski, *Nonlinear Optimization*. Princeton University Press, 2006.

[41] H. Xie, F. Gao, S. Jin, J. Fang, and Y. Liang, "Channel estimation for TDD/FDD massive MIMO systems with channel covariance computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4206–4218, 2018.

[42] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7485–7497, 2020.

[43] A. Li, D. Spano, J. Krivochiza, S. Domouchtsidis, C. G. Tsinos, C. Masouros, S. Chatzinotas, Y. Li, B. Vucetic, and B. Ottersten, "A tutorial on interference exploitation via symbol-level precoding: Overview, state-of-the-art and future directions," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 796–839, 2020.

[44] T. Lipp and S. Boyd, "Variations and extension of the convex - concave procedure," *Optimization & Engineering*, vol. 17, no. 2, pp. 263–287, 2016.

[45] W. Pu, Y. C. Eldar, and M. R. D. Rodrigues, "Optimization guarantees for ISTA and ADMM based unfolded networks," in *ICASSP 2022*, 2022, pp. 8687–8691.

[46] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5289–5304, Oct 2017.

Jianjun Zhang (Member, IEEE) received the M.S. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014, and the Ph.D. degree from Southeast University, Nanjing, China, in 2018.

Since July 2022, he has been a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From December 2019 to April 2022, he was a Research Fellow of the electrical and electronics engineering with University College London (UCL), London, UK. From March 2019 to November 2019, he was a Post-Doctoral Researcher with the Purple Mountain Laboratories, Nanjing, China. He was the recipient of the Best Paper Award in the IEEE Globecom 2019. His current research interests include machine learning, theory and algorithms of optimization, and wireless communications and networking.

Christos Masouros (Senior Member, IEEE) received the Diploma degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2004, and MSc by research and PhD in Electrical and Electronic Engineering from the University of Manchester, UK in 2006 and 2009 respectively. In 2008 he was a research intern at Philips Research Labs, UK. Between 2009-2010 he was a Research Associate in the University of Manchester and between 2010-2012 a Research Fellow at Queen's University Belfast. In 2012 he joined University College London as a Lecturer. He has held a Royal Academy of Engineering Research Fellowship between 2011-2016.

He is currently a Full Professor in the Information and Communications Engineering research group, Dept. Electrical and Electronic Engineering, University College London. His research interests lie in the field of wireless communications and signal processing with particular focus on Green Communications, Large Scale Antenna Systems, Cognitive Radio, interference mitigation techniques for MIMO and multicarrier communications. He was the recipient of the Best Paper Awards in the IEEE GlobeCom 2015 and IEEE WCNC 2019 conferences, and has been recognised as an Exemplary Editor for the IEEE Communications Letters, and as an Exemplary Reviewer for the IEEE Transactions on Communications. He is an Editor for IEEE Transactions on Communications, and IEEE Transactions on Wireless Communications. He has been an Associate Editor for IEEE Communications Letters, and a Guest Editor for IEEE Journal on Selected Topics in Signal Processing issues "Exploiting Interference towards Energy Efficient and Secure Wireless Communications" and "Hybrid Analog/Digital Signal Processing for Hardware-Efficient Large Scale Antenna Arrays". He is currently an elected member of the EURASIP SAT Committee on Signal Processing for Communications and Networking.

Lajos Hanzo (Life Fellow, IEEE) received the master's and Ph.D. degrees from the Technical University (TU) of Budapest in 1976 and 1983, respectively, the Doctor of Science (D.Sc.) degree from the University of Southampton in 2004, and the dual Honorary Doctorate degree from the TU of Budapest in 2009 and the University of Edinburgh in 2015.

Since 1986, he has been a member of academic staff at the School of Electronics and Computer Science, University of Southampton, U.K., where he is currently holds the Chair in Telecommunications. He is currently a Foreign Member of the Hungarian Academy of Sciences and a Former Editor-in-Chief of the IEEE Press. He has served several terms as a Governor of IEEE ComSoc and VTS. He has published more than 2000 contributions at IEEE Xplore, 19 Wiley-IEEE Press books, and has helped the fast-track career of 123 Ph.D. students. Over 40 of them are professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is also a fellow of the Royal Academy of Engineering (FREng), IET, and EURASIP. He was a recipient of the 2022 Eric Sumner Field Award. (http://www-mobile.ecs.soton.ac.uk, https://en.wikipedia.org/wiki/Lajos_Hanzo).