

Unsupervised Deep Unfolded PGD for Transmit Power Allocation in Wireless Systems

Ramoni Adeogun

AI for Communications Group, Wireless Communication Network Section

Department of Electronic Systems, Aalborg University, Denmark

E-mail: ra@es.aau.dk

Abstract—Transmit power control (TPC) is a key mechanism for managing interference, energy utilization, and connectivity in wireless systems. In this paper, we propose a simple low-complexity TPC algorithm based on the deep unfolding of the iterative projected gradient descent (PGD) algorithm into layers of a deep neural network and learning the step-size parameters. An unsupervised learning method with either online learning or offline pretraining is applied for optimizing the weights of the DNN. Performance evaluation in dense device-to-device (D2D) communication scenarios showed that the proposed method can achieve better performance than the iterative algorithm with more than a factor of 2 lower number of iterations.

Index Terms—Power control, machine learning, deep unfolding, PGD, device-to-device communication.

I. INTRODUCTION

Transmit power control (TPC) has been studied extensively as a viable radio resource management technique for optimizing the performance as well as energy consumption of wireless systems, particularly in interference-limited scenarios. TPC typically involves the maximization (or minimization) of a communication theoretic objective function which is typically non-convex and NP-hard making deriving an optimal solution to become more difficult as the network size grows. Moreover, solutions must be obtained in real-time to adapt to the varying wireless channels.

Due to the non-convexity and NP-hardness of power allocation, existing works have resulted in approximate solutions which rely on convex or geometric programming approximation of the problem. This has led to various approximate power allocation methods based on techniques such as particle swarm optimization [1], game theory [2], and weighted minimum mean square error (WMMSE) algorithm [3]. The applicability of these numerical or iterative solutions for real-time operation is, however, hindered by their high computation complexity, particularly for large-scale networks. Moreover, their performance is known to be sensitive to initialization and convergence may sometimes be too slow for real-time processing.

Consequently, academic and industrial research focus appears to have shifted to machine learning-based algorithms with several published works applying graph neural networks [4]–[6], deep neural networks (DNNs) [7], reinforcement learning [8], [9], spatial DNN [10] to the transmit power allocation problem in wireless networks. These learning-based

solutions have been shown to achieve reasonably good performance with much lower complexity than classical iterative or heuristics algorithms for TPC. However, they are either based on the unrealistic assumption that network data for training machine learning models are available or that simulators can be developed to accurately model real wireless environments. In the wireless communication domain, the availability of suitable real data sets is still a challenge. Also, most of the existing machine learning-based methods suffer from slow convergence, non-interpretability, and/or overly complex network architecture.

To circumvent these limitations, there has been an increased interest in the deep unfolding of iterative algorithms [11], which aims to leverage the expressiveness and training capabilities of deep neural networks (DNNs) to improve performance with lower complexity compared to iterative algorithms. This approach allows for incorporating existing expert knowledge into the solution and has been applied to a wide range of optimization problems, including massive MIMO detection [12], compressed sensing, sparse coding, and channel coding [13]. By unfolding the iterations of an iterative algorithm into a DNN, it becomes possible to optimize the algorithm through gradient descent, using backpropagation to adjust the DNN weights [14]. This approach has been shown to improve convergence rates, reduce the required number of iterations, and allow for the incorporation of additional constraints or objectives into the solution.

In this paper, we propose an unsupervised deep unfolded projected gradient descent (DUPGD) algorithm for power control in wireless networks. Our approach is based on deep unfolding the projected gradient descent algorithm, which is a widely used iterative algorithm for solving optimization problems. The DUPGD is trained following an unsupervised framework with the loss determined by the objective function in the TPC optimization problem. Performance evaluation results in device-to-device (D2D) communication scenarios have shown that DUPGD can achieve better performance than the classical iterative PGD for power allocation.

The rest of this paper is organized as follows. The system model as well as problem formulation is presented in Section II. In Section III, the iterative PGD algorithm and the proposed DUPGD methods are described. Performance evaluation results are presented and discussed in Section IV. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROPOSED ALGORITHMS

A. System Model

We consider a typical device-to-device communication scenario with N independent links (i.e., transmitter-receiver pairs) operating over a single shared frequency channel with bandwidth, B . The received signal-to-noise plus interference ratio (SINR) for the link between the n th transmitter and its receiver n' is defined as

$$\zeta_{nn'} = \frac{p_n h_{nn'}}{\sum_{m=1, m \neq n}^N p_m h_{mn'} + \sigma^2}, \quad (1)$$

where p_i denotes the transmit power of the transmitter, i , h_{ij} is the channel gain between transmitter i and receiver j' and σ^2 denotes the noise variance.

B. Problem Formulation

We consider the typical problem of maximizing the sum rate. The optimal transmit power vector $\mathbf{p} = [p_1, p_2, \dots, p_N]^T$ can then be obtained as the solution of

$$\begin{aligned} \text{P: } \hat{\mathbf{p}} &= \arg \min_{\mathbf{p} \in [0, p_{\max}]} - \sum_{n=1}^N \log_2(1 + \zeta_{nn'}) \\ \text{st: } &0 \leq p_n \leq p_{\max} \quad \forall n = 1, \dots, N \end{aligned} \quad (2)$$

where p_{\max} denotes the maximum transmit power per transmitter. The n th term in the summation corresponds to the rate achieved by the n th transceiver pair. We will henceforth use ρ to denote the objective function in (2).

III. PROPOSED METHOD

A. Iterative PGD for Power Allocation

To solve (2), the PGD algorithm starts with an initial guess of the transmit power vector, \mathbf{p}_0 , in the feasible set, i.e., $p_0 \in [0, p_{\max}]$ and computes the iterate

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - \lambda \nabla \rho, \quad k = 1, \dots, K \quad (3)$$

for a specified value of K or until an appropriate termination criterion is met. In (3), λ ($0 \leq \lambda < 1$) is the learning rate, and $\nabla \rho$ denotes the gradient of the negative sum rate ρ with respect to the transmit power vector, \mathbf{p} , i.e.,

$$\nabla \rho = \frac{\delta \rho}{\delta \mathbf{p}} = \begin{bmatrix} \frac{\delta \rho}{\delta p_1} & \frac{\delta \rho}{\delta p_2} & \dots & \frac{\delta \rho}{\delta p_N} \end{bmatrix}^T. \quad (4)$$

The derivative $\frac{\delta \rho}{\delta p_n}$ can be obtained using standard differential calculus rules to be

$$\frac{\delta \rho}{\delta p_n} = \frac{1}{N \log(2)} \left(\frac{h_{nn}}{\gamma_n} - \sum_{k=1, k \neq n}^N \frac{h_{kk} h_{kn} p_k}{(\eta_k)(\eta_k + h_{kk} p_k)} \right), \quad (5)$$

where $\gamma_i = \sum_{m=1}^N p_m h_{im} + \sigma^2$ and $\eta_i = \sum_{m=1, m \neq i}^N p_m h_{im} + \sigma^2$. In PGD, the iterate in (3) is followed by projection onto the non-negative orthant and onto p_{\max} . A summary of the iterative PGD algorithm for TPC is shown in Algorithm 1. Note that the convergence and the closeness of the power allocation in (3) to optimal solutions are highly dependent on the choice of the step-size parameter, λ .

Algorithm 1: Iterative PGD for TPC.

Input: Transmit powers \mathbf{p} , channel gains \mathbf{h} , noise variance σ^2 , learning rate α , number of iterations, K

Result: Optimal transmit powers \mathbf{p}^*

- 1 Initialize \mathbf{p} with some initial value;
- 2 **for** $k = 1, 2, \dots, K$ **do**
- 3 Compute the $\nabla \rho$ with respect to \mathbf{p} using (4);
- 4 Update the transmit powers using (3);
- 5 Project the transmit powers onto the feasible set:
 $\mathbf{p} = \text{proj}_{[0, P_{\max}]^N}(\mathbf{p});$
- 6 **end**
- 7 **return** $\mathbf{p}^* = \mathbf{p};$

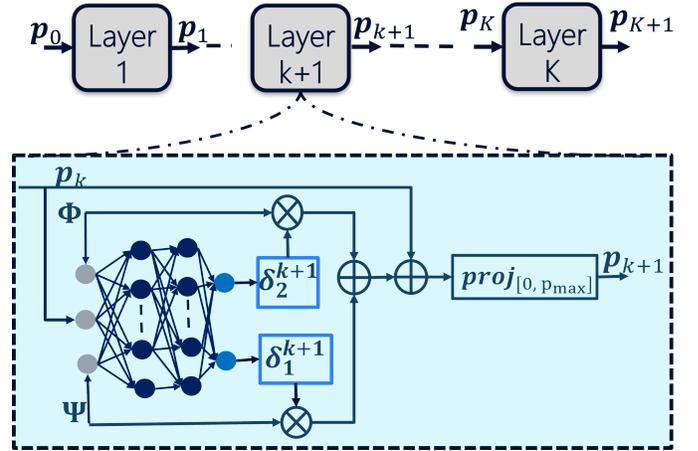


Fig. 1. Illustration of proposed deep unfolded PGD for wireless power control.

B. Deep Unfolded PGD for Power Allocation

Deep unfolding is a technique that involves transforming an iterative optimization algorithm into a deep neural network (DNN) by unfolding the iterations of the algorithm into layers of the DNN. The resulting DNN approximates the optimal solution of the iterative algorithm. We unfold the iterative PGD algorithm into a deep neural network (DNN) which can be trained offline translating to reduced execution complexity. By substituting (5) and (4) into the PGD iterate in (3), we obtain

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - \lambda \Psi + \lambda \Phi, \quad k = 1, \dots, K, \quad (6)$$

where

$$\Psi = \frac{1}{\log(2)} \begin{bmatrix} \frac{h_{11}}{\gamma_1}, \dots, \frac{h_{NN}}{\gamma_N} \end{bmatrix}^T \quad (7)$$

and

$$\Phi = \frac{1}{\log(2)} \begin{bmatrix} \sum_{k=1, k \neq 1}^N \frac{h_{kk} h_{k1} p_k}{(\eta_k)(\eta_k + h_{kk} p_k)}, \dots, \sum_{k=1, k \neq N}^N \frac{h_{kk} h_{kN} p_k}{(\eta_k)(\eta_k + h_{kk} p_k)} \end{bmatrix}^T. \quad (8)$$

TABLE I
DEFAULT SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Deployment area	20 m x 20 m	Number of links, N	20
Pathloss exponent	2	Shadowing standard deviation	5 dB
Step size, λ	0.1	Max number of iterations	1000
Number of DUPGD layers	40	Number of neurons/layer	16

As seen in (6), we require the terms \mathbf{p}_k , Φ and Ψ to perform the $k + 1$ -th iteration of the PGD algorithm for TPC. These terms are then used as the input to the proposed deep unfolded PGD (DUPGD) model as shown in Fig. 1. Denoting the learned value of $-\lambda$ and λ at iteration $k + 1$ as δ_1^{k+1} , δ_2^{k+1} , the operations performed at the corresponding layer of DUPGD can be written as

$$\mathbf{p}_{k+1} = \text{proj}_{[0, p_{\max}]} (\hat{\mathbf{p}}_k + \delta_1^{k+1} \Psi + \delta_2^{k+1} \Phi) \quad (9)$$

The overall structure of the DUPGD is depicted in Fig. 1. Unlike the iterative PGD method which relies on a manually specified value of the step-size parameter, λ , the step-size parameters are automatically determined in DUPGD via an unsupervised training procedure. In DUPGD, the step-size parameter is split into two learned parameters, $\delta_1^k \in [-1, 0]$ and $\delta_2^k \in [0, 1]$ translating to an increased degree of freedom. At each layer of the DUPGD, the channel matrix, \mathbf{h} and transmit power vector, \mathbf{p} are converted into the vectors, Φ and Ψ which are then passed as inputs to the layer as shown in Fig. 1.

1) *Training procedure*: The DUPGD model is trained following the unsupervised training method described in [11] with the loss function defined as the negative of the sum rate averaged over a batch of N_B instantaneous channel gain matrices. During training, the model is applied to predict transmit power levels for all nodes. The predicted transmit power is then used to update the weights of the DNN via backpropagation. In this, we used the Adam optimizer. It should, however, be noted that other stochastic gradient descent algorithms can be applied. We investigate two kinds of models viz: DUPGD with instantaneous online training (denoted DUPGD-online) and DUPGD with offline training (denoted DUPGD-offline), depending on how the training is performed. In DUPGD-online, the update of the model weights is performed layer-wise using a single instantaneous channel realization. In contrast, DUPGD-offline is trained prior to online execution with stochastic gradient descent over multiple batches of channel realizations. Once trained, the model is then deployed for online transmit power allocation translating to a lower execution complexity relative to DUPGD-online.

IV. PERFORMANCE EVALUATION

A. Simulation settings

We consider a D2D scenario with $N = 20$ links where the transmitters and receivers are randomly placed in a $20 \text{ m} \times 20 \text{ m}$ rectangular area. We consider deployments with no transmitter-to-receiver distance restriction and with

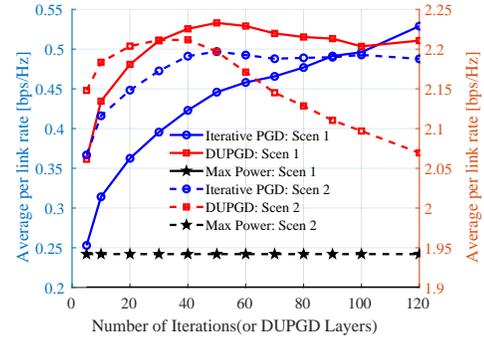


Fig. 2. Average per link rate versus number of iterations with $N = 10$.

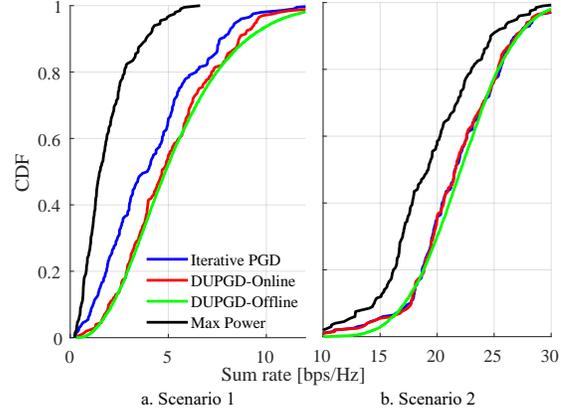


Fig. 3. CDF of sum rate with number of links, $N = 10$.

a maximum transmitter-to-receiver distance of $d_{\max} = 3 \text{ m}$ which are henceforth denoted Scen 1 and Scen 2, respectively. While Scen 1 corresponds to a typical D2D communication scenario, Scen 2 represents scenarios where a transmitter communicates with a receiver in close proximity as in, e.g., short-range low-power 6G in-X subnetworks [15], [16] for extreme wireless communication inside entities such as the human body, industrial robots, and vehicles. Transmission bandwidth of $B = 5 \text{ MHz}$ with carrier frequency, $f_c = 6 \text{ GHz}$ is considered in the simulations. With a noise figure of 10 dB, this translate to a noise power of $\sigma^2 = 2 \times 10^{-10}$. Other simulation parameters are presented in Table I. Except where stated otherwise, the maximum number of iterations for the iterative PGD method is set to 1000. In the simulations, each layer of the DUPGD consists of a DNN with two hidden layers with 64 neurons per layer. Each hidden layer is followed by a rectified linear unit (ReLU) activation function. At the output layer, a clamping operation is performed to project the DNN prediction onto the feasible range between 0 and p_{\max} .

B. Channel model

The channel gain, h is computed using

$$h_{ij'} = g_{ij'} \left(\frac{c^2 \min(1, d_{ij'}^{-\alpha})}{16\pi^2 f_c^2} \Xi_{ij'} \right)^{\frac{1}{2}}, \quad (10)$$

where $g_{ij'}$ is small-scale fading gain generated as Rayleigh distributed random variables, $d_{ij'}$ is the distance between

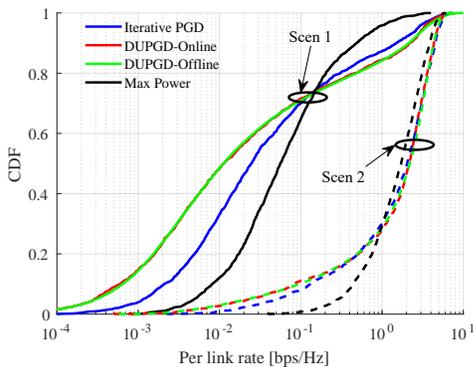


Fig. 4. CDF of per link rate with number of links, $N = 10$.

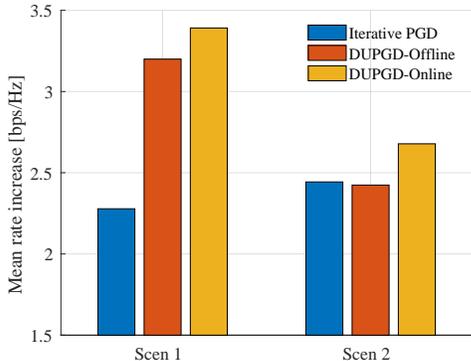


Fig. 5. Mean rate improvement relative to the maximum power baseline with $N = 10$ links.

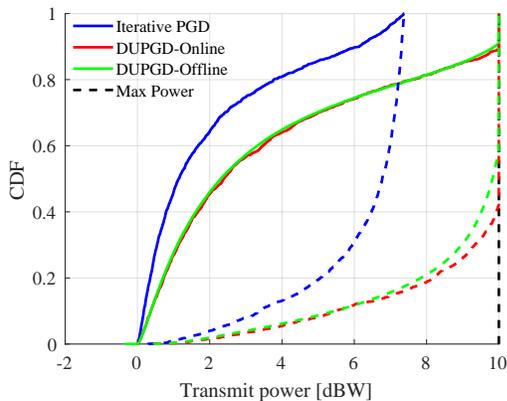


Fig. 6. Distribution of allocated transmit power.

transmitter i and receiver j' , and ω denotes the path-loss exponent. f_c , c , and $\Xi_{ij'}$ are respectively the carrier frequency, speed of light, and lognormal distributed shadowing of the link between the transmitter, i , and receiver j' .

C. Evaluation results

We evaluate the performance of the proposed DUPGD algorithm and compare it with that of classical iterative PGD and maximum power. In Fig. 2, we show the average per link rate as a function of the number of iterations (or equivalently the number of layers in DUPGD). As shown in the figure the average per-link rate (or equivalently the sum rate) increases

with the number of iterations for both the DUPGD and iterative PGD. For the DUPGD methods, the maximum rate is achieved with the number of layers of approximately 40. This indicates that a maximum of 40 DUPGD layers are sufficient to achieve the best performance in the scenarios considered. In contrast, it appears that the iterative PGD's average rate performance remains constant after 60 iterations for Scen 2 and continues to increase for Scen 1. It is therefore reasonable to conclude that the DUPGD requires a much less number of layers than the number of iterations in the iterative PGD. To minimize the effect of early termination on the performance achieved by the iterative PGD algorithm, we henceforth set the maximum number of iterations to 1000. Fig. 3 shows the CDF of the sum rate achieved by the proposed and benchmark methods in scenario 1 (a) and scenario 2 (b). The figure shows that the DUPGD performs better than the equal power baseline in both scenarios and is similar in performance to the complex iterative PGD algorithm in scenario 2. In scenario 1, which is more challenging due to the completely random location of the transmitters and receivers, the DUPGD algorithm yields slightly better performance. The figure further shows no significant difference between the performance of DUPGD with offline and online training. This indicates that the proposed method can be used without any pre-training in cases where computation complexity associated with online training can be tolerated. In Fig. 4, we show the distribution of per link rate for the different algorithms. The results indicate that maximization of the sum rate as in iterative PGD and DUPGD leads to enhanced individual rates at the higher percentile of the CDF while penalizing the lower percentiles. However, this is expected considering that the optimization focuses only on the sum rate. In cases with a minimum rate requirement, the objective function can be augmented with a penalty term defined in terms of such a requirement.

To quantify the relative performance improvement, we show the mean rate increase computed as the difference between the achieved rate by a specific algorithm and that achieved by the maximum power baseline in Fig. 5. The figure shows that the proposed methods yield significantly higher mean rate increases of 3.39 bps/Hz (3.19 bps/Hz) for DUPGD-Online (Offline) compared to iterative PGD with an increase of 2.27 bps/Hz in scenario 1. This shows that the proposed methods can achieve up to $\approx 49\%$ improvement in the relative rate increase compared to maximum power. The performance improvement in scenario 2 is only about 9.43% (for DUPGD-Online) which is much lower than that achieved in scenario 1. A plausible explanation for this observation is that the relatively strong desired link power (due to the distance constraint) diminishes the impact of power control on the experienced aggregate interference level at each node.

In Fig. 6, we show the distribution of the power allocated to each transmitter by the different methods. The figure shows that the iterative algorithm results in lower power consumption than the DUPGD in the considered scenarios. For instance, while the no transmitter gets the maximum power with the iterative algorithm, approximately 10% and 60% are allocated

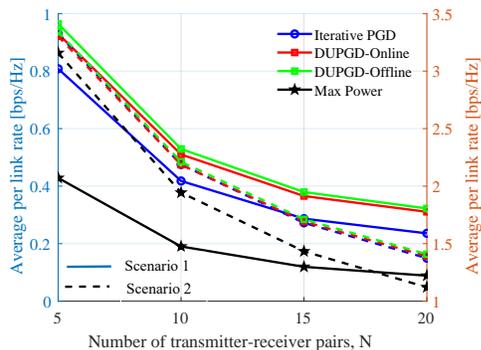


Fig. 7. Average rate versus the number of links.

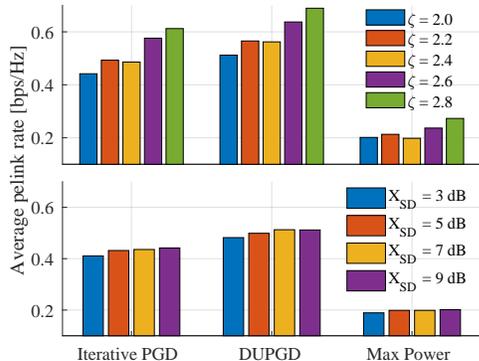


Fig. 8. Sensitivity of TPC algorithms to changes in path loss and shadowing.

the maximum power of 10 dBW with DUPGD in scenarios 1 and 2, respectively. The mean transmit power with iterative PGD and DUPGD are 2.69 dBW and 5.21 dBW (in scenario 1) and 6.31 dBW and 9.27 dBW (in scenario 2).

D. Sensitivity evaluation

We study the sensitivity of the proposed DUPGD method to changes in deployment density (or equivalently the number of subnetworks with fixed deployment area) and propagation conditions. In Fig. 7, we show the average per link rate as a function of the number of transmitter-receiver pairs, N , in a $20 \text{ m} \times 20 \text{ m}$ area. The figure shows that the relative performance of the proposed DUPGD to the baseline iterative algorithm remains constant for all values of N . This indicates that the DUPGD generalizes well to different deployment densities. Fig. 8 shows the sensitivity of the power allocation methods to changes in the path-loss exponent and shadowing standard deviation. The figure shows consistent relative performance between the DUPGD method and the benchmark algorithms for all values of path-loss exponent and shadowing standard deviation. It is therefore reasonable to conclude that the proposed method is robust to changes in wireless environment conditions.

V. CONCLUSION

We proposed a low-complexity power control algorithm based on a deep unfolding of iterative project gradient descent. We showed via extensive simulations in device-to-device communication scenarios that the proposed method can achieve

better performance than the classical iterative algorithm and decreases the number of iterations by more than a factor of 2. Simulation results have shown that the proposed algorithm exhibits similar performance relative to the benchmark methods in environments with different propagation characteristics and deployment densities.

ACKNOWLEDGEMENT

This research is partially supported by the HORIZONJUSNS-2022-STREAM-B-01-03 6G-SHINE project (Grant Agreement No. 101095738).

REFERENCES

- [1] M. Khodier and G. Saleh, "Beamforming and power control for interference reduction in wireless communications using particle swarm optimization," *AEU-International Journal of Electronics and Communications*, vol. 64, no. 6, pp. 489–502, 2010.
- [2] S. Sengupta, M. Chatterjee, and K. Kwiat, "A game theoretic framework for power control in wireless sensor networks," *IEEE transactions on computers*, vol. 59, no. 2, pp. 231–242, 2009.
- [3] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.
- [4] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.
- [5] D. O. Abode, R. O. Adeogun, and G. Berardinelli, "Power control for 6g industrial wireless subnetworks: A graph neural network approach," in *IEEE Wireless Communications and Networking Conference. Proceedings*. IEEE Communications Society, 2023.
- [6] Y. Gu, C. She, Z. Quan, C. Qiu, and X. Xu, "Graph neural networks for distributed power allocation in wireless networks: Aggregation over-the-air," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.
- [7] I. Hameed, P. V. Tuan, and I. Koo, "Deep learning-based energy beamforming with transmit power control in wireless powered communication networks," *IEEE Access*, vol. 9, pp. 142 795–142 803, 2021.
- [8] Y. Lu, H. Lu, L. Cao, F. Wu, and D. Zhu, "Learning deterministic policy with target for power control in wireless networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.
- [9] R. Zhang, K. Xiong, W. Guo, X. Yang, P. Fan, and K. B. Letaief, "Q-learning-based adaptive power control in wireless rf energy harvesting heterogeneous networks," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1861–1872, 2020.
- [10] T. Zhang and S. Mao, "Energy-efficient power control in wireless networks with spatial deep neural networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 111–124, 2019.
- [11] S. He, S. Xiong, Z. An, W. Zhang, Y. Huang, and Y. Zhang, "An unsupervised deep unrolling framework for constrained optimization problems in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8552–8564, 2022.
- [12] S. Takabe, M. Imanishi, T. Wadayama, R. Hayakawa, and K. Hayashi, "Trainable projected gradient detector for massive overloaded mimo channels: Data-driven tuning approach," *IEEE Access*, vol. 7, pp. 93 326–93 338, 2019.
- [13] T. Wadayama and S. Takabe, "Deep learning-aided trainable projected gradient decoding for ldpc codes," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 2444–2448.
- [14] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [15] R. Adeogun, G. Berardinelli, and P. E. Mogensen, "Enhanced interference management for 6g in-x subnetworks," *IEEE Access*, vol. 10, pp. 45 784–45 798, 2022.
- [16] R. Adeogun, G. Berardinelli, P. E. Mogensen, I. Rodriguez, and M. Razzaghpour, "Towards 6g in-x subnetworks with sub-millisecond communication cycles and extreme reliability," *IEEE Access*, vol. 8, pp. 110 172–110 188, 2020.