

Learning-Based Antenna Selection for Multicasting

Mohamed S. Ibrahim*, Ahmed S. Zamzam*, Xiao Fu[†] and Nicholas D. Sidiropoulos[‡]

*Dept. of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN

[†]School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR

[‡]Dept. of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA

Email: {{youss037, ahmedz}@umn.edu, xiao.fu@oregonstate.edu, nikos@virginia.edu}

Abstract—In multi-antenna systems, it is preferred to activate only a subset of the available transmit antennas in order to save hardware and energy resources, without seriously degrading the system performance. However, antenna selection often poses very hard optimization problems. Joint multicast beamforming and antenna selection is one particular example, which is often approached by Semi-Definite Relaxation (SDR) type approximations. The drawback is that SDR lifts the problem to a much higher dimension, leading to considerably high memory and computational complexities. In this paper, we propose a machine learning based approach to circumvent the complexity issues. Specifically, we propose a neural network-based approach that aims at selecting a subset of antennas that maximizes the minimum signal to noise ratio at the receivers. The idea is to learn a mapping function (represented by a neural network) that maps channel realizations to antenna selection solutions from massive simulated data. This way, the computational burden of antenna selection can be shifted to off-line neural network training. Experiments demonstrate the efficacy of the proposed machine learning approach relative to the prior state-of-art.

I. INTRODUCTION

In wireless communications, transmit beamforming is an effective technique for increasing the capacity in systems in which the transmitter is equipped with an antenna array. However, it is considered economically inefficient to have the same number of radio transmission chains as the antennas, owing to the cost, power and size considerations. One way to reduce the cost while keeping the diversity/beamforming gain advantage of multiple antennas is antenna selection; i.e., operating with an appropriate subset of the available antennas.

Consider a downlink scenario where a multi-antenna transmitter sends common messages to multiple single antenna receivers. The multicast beamforming problem exploits channel state information in order to enhance the subscribers' quality of service. This setup was initially introduced in [1] where it was shown that the problem is NP-hard. The hardness carries over to the joint antenna selection and beamforming problem that was first formulated in [2]. Semidefinite relaxation (SDR)-based approaches have been shown effective in identifying near-optimal solutions for both problems at the price of high computational and memory complexities [1], [2]. The SDR procedure lifts the optimization task into an equivalent higher dimensional problem with rank constraints. Then, the problem is relaxed by dropping the rank constraint.

The solution obtained by solving the relaxed reformulation is not guaranteed to be feasible for the original problem. Therefore, the approach is followed by a randomization step which adds to the computational burden of the algorithm.

On the other hand, the use of Machine Learning (ML) and Deep Neural Networks (DNNs) in wireless communications has recently attracted a lot of attention [3]–[5]. For instance, the authors in [3] proposed a multi-class classification approach to tackle the antenna selection problem considered in [6]. In order to increase the spatial multiplexing gain of the system, multiclass k-nearest neighbors and support vector machine approaches were employed. Known as universal function approximators, neural networks have recently made a remarkable comeback, outperforming far more disciplined methods in numerous applications; see [7]–[9] for examples. One appealing feature of using DNNs (and other machine learning approaches) for antenna selection is that the computational burden is now shifted to the off-line training stage. Once the DNN's coefficients are tuned, the use of the DNN requires only simple computations, thus bringing real-time implementation for complicated tasks within reach. This idea was considered for power allocation in transmit signal design [4], [5], and impressive performance has been observed.

In this work, we propose a DNN-based approach to handle the joint multicast beamforming and antenna selection problem, to alleviate the computational burden of the base stations when performing this very hard task. Using a deep neural network, the proposed approach aims at learning a nonlinear mapping between the channel state information and a *good* antenna selection in terms of overall system performance. A natural thought of approaching this problem is to train a neural network that directly maps the channel state information to a set of selected antennas with optimized beamforming weights. However, since the joint antenna selection and beamforming problem (which is a mixed optimization problem with discrete and continuous variables) is intrinsically hard, this attempt often times fails yield reasonable results. To overcome the difficulties, an innovative training approach is proposed. First, instead of taking the channel state information as the input of the neural network, the second-order correlation (information) of the channel are used as input. The intuition is that only second-order information matters in this problem and feeding this information directly as input saves network resources and the learning needed to extract the most relevant information

This work is supported by a Graduate Fellowship awarded by the Digital Technology Center (DTC) at the University of Minnesota.

from the raw input. In addition, we fix the DNN's output to be a vector with binary elements, indicating whether or not the corresponding antennas are selected, and leave out the beamforming part to a follow-up stage. We find that this training approach is empirically much more effective compared to training a network that performs joint antenna selection and beamforming. Once the set of selected antennas is observed at the DNN output, an efficient first order-based method based on the Mirror-Prox (MP) algorithm is utilized to obtain the corresponding beamforming weights [10]. Numerical experiments using Gaussian random channels corroborate the superior performance of the proposed method over existing state-of-the-art algorithms.

Regarding notation, matrices (vectors) are denoted by upper- (lower-) case boldface letters, and $(\cdot)^T$, and $(\cdot)^H$ stand for transpose, and conjugate-transpose, respectively. $\|\cdot\|_2$, $\|\cdot\|_1$, and $\|\cdot\|_0$ denote the ℓ_2 -, ℓ_1 -, and the cardinality of (\cdot) , respectively.

II. PRELIMINARIES

A. System Model

Consider a scenario where a single base station (BS) equipped with N transmit antennas broadcasts a common message to M single-antenna receivers. Let \mathbf{h}_m denote the $N \times 1$ complex vector that models the propagation loss and the frequency-flat quasi-static channel from the transmitter to user m , where $m \in \{1, \dots, M\}$. In addition, the beamforming weight vector applied at the BS is denoted by $\mathbf{w} \in \mathbb{C}^N$. Hence, the received signal by the m -th receiver is given by

$$y_m = \mathbf{h}_m^H \mathbf{w} s + z_m,$$

where z_m is zero mean Gaussian noise with variance σ_m^2 . Assuming the signal to be transmitted is zero mean with unit variance, the received SNR at the m -th user can be expressed as

$$\frac{|\mathbf{h}_m^H \mathbf{w}|^2}{\sigma_m^2} = \mathbf{w}^H \mathbf{A}_m \mathbf{w},$$

where $\mathbf{A}_m := \frac{\mathbf{h}_m \mathbf{h}_m^H}{\sigma_m^2} \succeq \mathbf{0}$, $\forall m \in \{1, \dots, M\}$. It is further assumed that the channel vectors $\{\mathbf{h}_i\}_{i=1}^M$ and their respective noise variances $\{\sigma_i^2\}_{i=1}^M$ are known at the BS.

Let K denote the number of available RF chains at the BS, where $K \leq N$. The goal is to select the best K out of N antennas, and find the optimal beamforming vector \mathbf{w} so that the minimum received SNR is maximized, subject to a limited transmit power budget. The problem can be mathematically posed as follows

$$\max_{\mathbf{w} \in \mathbb{C}^N} \min_m \mathbf{w}^H \mathbf{A}_m \mathbf{w} \quad (1a)$$

$$\text{s. to } \|\mathbf{w}\|_2^2 \leq P, \quad \|\mathbf{w}\|_0 \leq K, \quad (1b)$$

where P represents the available power at the BS, and the second constraint implies that only K antennas are active at most. The constraint on the cardinality of \mathbf{w} is known to be non-convex, and hence renders the problem even harder to

solve (even with $K = N$, the problem is nonconvex and NP-hard for $M \geq N$).

One workaround is to penalize the one-norm of the beamforming vector

$$\max_{\mathbf{w} \in \mathbb{C}^N} \left\{ \min_m \mathbf{w}^H \mathbf{A}_m \mathbf{w} \right\} - \lambda \|\mathbf{w}\|_1 \quad (2a)$$

$$\text{s. to } \|\mathbf{w}\|_2^2 \leq P, \quad (2b)$$

where λ is a positive real tuning parameter that controls the sparsity of the solution, and hence, the number of selected antennas. Problem (2) is known to be NP-hard; however, high-quality solutions can be obtained using various algorithms [2], [11]. Specifically, the solution obtained in [2] is based on SDR techniques, while the one in [11] is obtained using the Mirror-Prox (MP) algorithm which is a first-order method introduced in [12].

Regardless of how successful these approaches prove to be in yielding high quality solutions, it remains very challenging to implement any of them in real time. This is mainly because they require bisection search for the optimal value of λ leading to the needed sparsity level. Afterwards, a reduced size problem with the set of selected antennas can be solved to get the optimal beamforming vector. In other words, the main issue that increases the computational time for both the SDR and Mirror-Prox type algorithms is the antenna selection part – as searching for the appropriate λ requires running the algorithms many times. In what follows, we first review the MP algorithm which will also be employed to generate the training data for our learning based approach.

B. Mirror-Prox Algorithm

The joint multicast beamforming and antenna selection problem is difficult because it is a non-smooth, non-convex optimization problem. However, there exists a general approximation framework, namely, Successive Convex Approximation (SCA) [13], [14] that can be applied to obtain approximate solutions. SCA approximates the original non-convex problem by a sequence of convex problems. The algorithm is initialized from a feasible point $\mathbf{w}^{(0)}$ and proceeds as follows. In every iteration, a convex surrogate of (2) is constructed by locally linearizing each quadratic component of the objective. Then, the algorithm proceeds via successively solving a series of such local convex approximations. This approach is used in [10] to solve the max-min multicast beamforming, where the local surrogate problem in the n th iteration is as follows

$$\max_{\mathbf{w} \in \mathbb{C}^N} \min_m \left\{ \mathbf{a}_m^{(n)T} \mathbf{w} + \mathbf{b}_m^{(n)} \right\}_{m=1}^M \quad (3)$$

$$\text{s. to } \|\mathbf{w}\|_2^2 \leq P$$

where $\mathbf{a}_m^{(n)} := 2\mathbf{A}_m \mathbf{w}^{(n)}$ and $\mathbf{b}_m^{(n)} := \mathbf{w}^{(n)H} \mathbf{A}_m \mathbf{w}^{(n)}$.

Since (3) does not admit a closed form solution, iterative methods must be utilized to obtain approximate solutions of each SCA problem. In particular, [10] used a modified version of the Alternating Direction Method of Multipliers algorithm (ADMM) to solve (3). In addition to ADMM, [10] also considered the iterative MP algorithm which is based

on the mirror descent procedure [15] to solve (3). Following the same approach, we used the concepts of group sparsity and dual norm [16], [17] to formulate the problem of joint multicast beamforming and antenna selection such that it can be solved using the MP method [11]. The reason behind adopting the fast MP algorithm to solve each SCA is that the method exhibits dimension independent $O(1/t)$ -rate of convergence. Although the simulations in [11] showed that MP provides high quality solutions to (2), there is still a considerable gap from real-time implementation.

III. PROPOSED APPROACH

In this section, a two-stage approach is proposed for efficiently selecting the set of active antennas as well as finding the respective beamforming vector that achieves the max-min fairness design goal. First, a DNN-based approach is devised to approximate a mapping from channel realizations to antenna selection solutions. Upon careful selection of the set of active antennas, the MP algorithm is utilized to design the beamforming weights by solving a reduced size problem. The aim is to leverage the computational efficacy of the DNN in order to obtain a fast real time antenna selection scheme. Therefore, the joint beamformer design and antenna selection problem is rendered more amenable to first-order approaches.

A. DNN Settings

The neural network used consists of one input layer, two hidden layers, and one output layer, as Fig. 1 depicts. Since the second order information of the channel $\{\mathbf{h}_m \mathbf{h}_m^H\}_{m=1}^M$ is actually what is used as input in (1) to determine the appropriate set of active antennas, our idea is to use these outer products instead of the channel vectors *per se* as input to the neural network. While this may *a posteriori* seem intuitive, it is not *a priori* obvious, because expanding the input size may also seem to lead to unnecessary over-parametrization. Can such input expansion and over-parametrization lead to tangible benefits in practice? As we will see, it does. In our extensive simulations, we found that using second-order information as the input improves the performance of the system substantially. Since \mathbf{A}_m is a hermitian matrix, the network is fed with the real and imaginary parts of the upper triangular part of the matrices $\{\mathbf{A}_m\}_{m=1}^M$. The output of the network is a vector of length N that represents which antennas are activated.

B. Training Samples Generation

The DNN is trained using pairs of channel realizations and binary antenna selection patterns. A total of L instances are used to train the network. Since the second order information of the channels is used as input to the network, the number of nodes at the input layer is given by MN^2 . Thus, any training sample is realized in the following way. First, the channel realizations $\{\mathbf{h}_j\}_{j=1}^M$ are generated following an appropriate channel model distribution. Then, the second order information matrices $\{\mathbf{A}_m\}_{m=1}^M$ are constructed. Since the matrices are hermitian, we can build the input vector \mathbf{x}_i by vectorizing

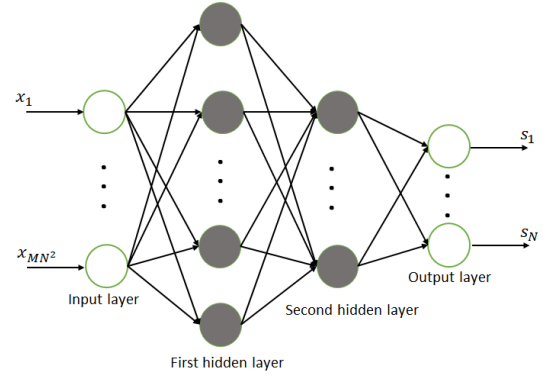


Fig. 1: DNN structure

all the real and imaginary parts of the upper triangular parts of the matrices \mathbf{A}_m 's. Then, for each channel realization, the MP algorithm is used to decide which K out of N antennas can achieve max-min fairness under the power budget constraint in (3). Afterwards, the target vector \mathbf{s} is designed by assigning ones to the set of K chosen antennas and zeros to the others. Now, the i -th training sample contains the tuple $(\mathbf{x}_i, \mathbf{s}_i)$, where $i \in \{1, \dots, L\}$. This process is repeated to generate the entire training data set and the validation data set which is used for cross validation. The training data matrix \mathbf{X} is constructed by stacking $\{\mathbf{x}_i\}_{i=1}^L$ as $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_L]$. Similarly, the target matrix \mathbf{S} can be obtained by stacking the binary vectors \mathbf{s}_i 's as $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_L]$.

Note that the proposed training strategy only maps \mathbf{A}_m to a binary vector, whose element indicate whether or not the corresponding antennas are selected. Compared to mapping the channel state information to as sparse \mathbf{w} (i.e., a vector with both the antenna selection information and the beamforming weights), the proposed training approach alleviates the 'learning burden' of the network—which makes the network much easier to train and leads to much better results in practice.

C. Training Stage

A real-valued matrix \mathbf{X} and a corresponding target matrix \mathbf{S} are constructed, where the i th columns of the matrices \mathbf{X} and \mathbf{S} represent the i th training sample and its corresponding target vector, respectively. Using the L training samples, the DNN tries to find the optimal weights that minimize a loss function given by

$$\ell(\mathbf{X}, \mathbf{S}) = \frac{1}{2L} \sum_{i=1}^L \|\mathbf{f}(\mathbf{x}_i) - \mathbf{s}_i\|_2^2$$

where $\mathbf{f}(\mathbf{x}_i)$ is the output of the neural network when the input is \mathbf{x}_i . The stochastic gradient descent algorithm is used for tuning the network parameters in order to minimize the loss function. The cross validation set is used for early stopping. The number of neurons in the first hidden layer is chosen to be twice the number of the inputs, while in the second layer, the number is chosen to be equal to the input size.

D. Testing Stage

In the testing stage, the channel realizations are generated following the same distribution as the training stage. Then, the upper triangular parts of the matrices $\{\mathbf{A}_m\}_{m=1}^M$ are obtained and fed to the neural network. Since the network has soft outputs between 0 and 1, the largest K values are considered to be the set of selected antennas. Afterwards, the MP-SCA algorithm, described in Section II-B, is utilized to solve a reduced-size problem with the set of selected antennas to get the corresponding beamforming vector that attains the max-min SNR.

IV. NUMERICAL RESULTS

In the simulations, a BS having $N = 6$ antennas broadcasting a common message to $M = 10$ receivers is used. The downlink channels $\{\mathbf{h}_m\}_{m=1}^M$ are modeled as random vectors drawn from a complex, circularly symmetric, normal distribution with zero mean and identity covariance matrix and the noise variance was set to be 1 for all the users. The total transmission power P was set to 10. The SCA algorithm was run for 15 iterations, with 1000 iterations being used to solve each SCA subproblem using the MP algorithm. All experiments were carried out on a Linux machine with Intel-i7 quad-core processor equipped with 8 GB of RAM.

To test the proposed DNN-based algorithm for antenna selection, the Neural Network toolbox for MATLAB was used to implement a two-layer neural network. The activation function of both layers was the Log-Sigmoid. The first layer consists of 720 neurons, while the second one contains 360 neurons. The size of the input is given by MN^2 , which is the vectorized upper triangular part of the second order information of the matrices \mathbf{A}_m , for $m \in \{1, \dots, M\}$. The output is an $N \times 1$ vector that represents the probability that antenna j is active, where $j \in \{1, \dots, N\}$. In the training stage, a total of 30000 channel realizations were used, where 3% of them are used for validation. Finally, the learning rate was set to 0.01. The number of selected antennas K is varied from 2 to 5. Then, the performance of the proposed method is benchmarked against four different algorithms:

- **Mirror prox:** the algorithm aspires to find the optimal beamformer that uses only K antennas by solving (2) for different values of λ .
- **Exhaustive Search:** in this algorithm, the multicast beamforming problem is solved for each of the combinations of K antennas—which is very expensive but with good performance.
- **The algorithm in [2]:** this is a SDR-based algorithm followed by a randomization technique to obtain the solution of (2).
- **Random Selection (RS):** The performance of this method is presented in order to show that by careful design, the system performance can be substantially improved compared to a random antenna selection strategy.

For each value of K , the performance of the DNN was assessed by choosing the K highest outputs from the network to be the set of selected antennas. Then, a reduced size

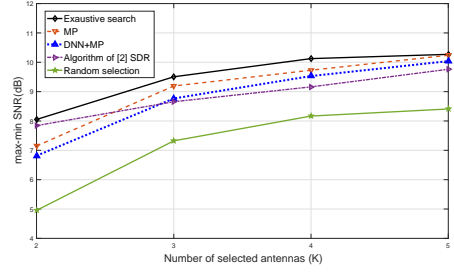


Fig. 2: The max-min SNR vs K

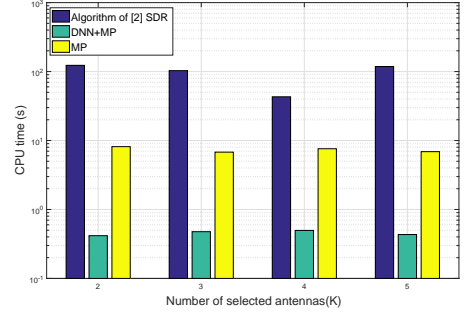


Fig. 3: CPU Time vs K

problem is solved using the MP algorithm to get the max-min SNR. Finally, the obtained solution is compared against the other four algorithms described above, where the results are averaged over 500 Monte-Carlo channel realizations. The results are depicted in Fig. 2 where the max-min SNR is plotted versus the number of selected antennas for each of the five algorithms. The figure shows that the quality of solutions obtained via the neural network outperforms that achieved using the algorithm in [2] except at the value of $K = 2$. It is worth noting that, in this scenario the performance of [2] is better than the MP algorithm which we use to obtain the beamformer after selecting the antennas.

Fig. 3 shows the runtime performance of the approaches. One can see that the proposed method (i.e. DNN+MP) clearly outperforms the optimization based methods. Specifically, the proposed method offers up to two orders of magnitude faster runtime performance relative to SDR, and is 15 times faster than solely using MP.

V. CONCLUSIONS

The problem of joint antenna selection and multicast beamforming was studied. The objective is to select the optimal set of antennas and identify the corresponding optimal beamforming vector that maximizes the minimum received SNR at the receivers. A DNN-based approach was put forth which tries to map the channel realization to the optimal antenna selection. Simulations demonstrated that the proposed DNN procedure provides substantial computational relative to traditional methods that solely rely on numerical optimization. Moreover, the combination of DNN-based initialization and optimization-

based refinement exhibits highly competitive performance in terms of providing max-min SNRs.

REFERENCES

- [1] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2239–2251, 2006.
- [2] O. Mehanna, N. D. Sidiropoulos, and G. B. Giannakis, "Multicast beamforming with antenna selection," in *IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Turkey, June 2012, pp. 70–74.
- [3] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2241–2244, 2016.
- [4] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled hetnet," in *International Symposium on Wireless Communication Systems (ISWCS)*, Italy, Aug. 2017, pp. 449–453.
- [5] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Signal Processing Advances in Wireless Communications (SPAWC)*, Japan, July 2017, pp. 247–252.
- [6] R. Heath and A. Paulraj, "Antenna selection for spatial multiplexing systems based on minimum error rate," in *IEEE International Conference on Communications (ICC)*, Finland, June 2001, pp. 2276–2280.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.
- [9] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.
- [10] A. Konar and N. D. Sidiropoulos, "Fast approximation algorithms for a class of non convex problems using first-order methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3494–3509, 2017.
- [11] M. S. Ibrahim, A. Konar, M. Hong, and N. D. Sidiropoulos, "Mirror-prox SCA algorithm for multicast beamforming and antenna selection," *arXiv preprint arXiv:1803.00678*, 2018.
- [12] A. Nemirovski, "Prox-method with rate of convergence $\mathcal{O}(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM J. Optim.*, vol. 15, no. 1, pp. 229–251, 2004.
- [13] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *Journal of Global Optimization*, vol. 47, no. 1, pp. 29–51, 2010.
- [14] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.
- [15] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3–4, pp. 231–357, 2015.
- [16] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [17] F. Bach and R. Jenatton, *Convex Optimization with Sparsity-Inducing Norms*.