

# Unsupervised Learning for Cellular Power Control

Rasoul Nikbakht, *Student Member, IEEE*, Anders Jonsson, Angel Lozano, *Fellow, IEEE*

**Abstract**—This paper applies a feedforward neural network trained in an unsupervised fashion to the problem of optimizing the transmit powers in cellular wireless systems. Both uplink and downlink are considered, with either centralized or distributed power control. Various objectives are entertained, all of them such that the problem can be cast in convex form. The performance of the proposed procedure is very satisfactory and, in terms of computational cost, the scalability with the system dimensionality is markedly superior to that of convex solvers. Moreover, the optimization relies on directly measurable channel gains, with no need for user location information.

**Index Terms**—Machine learning, neural networks, unsupervised learning, power control, cellular systems

## I. INTRODUCTION

A preceding letter proposed leveraging the expressive power of neural networks (NNs) to solve large-dimensional parametric optimizations [?]. In this follow-up letter, we apply this approach to the all-important problem of power control in cellular systems. In both uplink and downlink, power control is instrumental to prevent huge performance disparities among users. By defining suitable loss functions, it is possible to optimize the transmit powers so as to balance the user signal-to-interference-plus-noise ratios (SINRs) as desired. The more balanced the SINRs, the more fairness—customarily at the expense of a reduced average performance.

The use of NNs for cellular power control has been explored under supervised, semi-supervised, or outright unsupervised learning, with promising performance [?], [?], [?], [?], [?]. The unsupervised framework avoids the taxing precomputation of labeled training data for every conceivable environment and speeds up the retraining whenever there are changes in the system.

A relevant attribute of the supervised power control in [?] is that the input parameters are the user locations, from which a mapping is established to the transmit powers. Since location information could—especially indoors—be unreliable or outright unavailable, our designated parameters are the channel gains, which can be measured directly and reliably. However, in contrast with [?], [?], [?], we desire for the power control to respond only to the *large-scale* behavior of the channel gains, i.e., to their local-average values, and not to the *small-scale* fluctuations that occur on scales of milliseconds (in time) and hundreds of kilohertz (in frequency). This ensures relatively stable transmit powers, leaving the small-scale swings to be dealt with by link adaptation [?].

R. Nikbakht, A. Jonsson, and A. Lozano are with Univ. Pompeu Fabra, 08018 Barcelona (e-mail: angel.lozano@upf.edu). This work was supported by the European Research Council under the H2020 Framework Programme/ERC grant agreement 694974, by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502) as well as by MINECO's Projects RTI2018-102112 and RTI2018-101040, and by the ICREA Academia Program.

## II. CELLULAR SYSTEM MODEL

In its basic abstraction, a cellular system consists of  $N$  cells, each served by an AP. On a given time-frequency signaling resource, a single user is active within each cell. Both APs and users are equipped with an omnidirectional antenna.

The formulation in the sequel applies regardless of how the APs are positioned and, to be on the same footing as the cell-free systems considered in [?], we draw those positions uniformly at random. The Voronoi region of each AP then spawns a cell, and one user is dropped uniformly at random therein. Every user communicates with its same-cell AP while interfering with the rest of APs.

### A. Channel Features

Signals are subject to distance-dependent power decay with exponent  $\eta = 3.8$ , and this decay gives rise to a large-scale channel gain  $G_{n,k}$  between the  $k$ th user and the  $n$ th AP. Correspondingly, that link's signal-to-noise ratio is

$$\text{SNR}_{n,k} = \frac{G_{n,k}P}{\sigma^2} \quad (1)$$

with  $P$  the maximum transmit power and  $\sigma^2$  the noise power.

Recent results have shown that shadowing renders cellular systems approximately Poisson-like from the vantage of any user [?], a behavior that sharpens as the shadowing intensifies, but that is precise already for shadowing strengths of interest [?]. Relying on these findings, APs and users are positioned uniformly at random, with shadowing implicitly embedded in that randomness.

For the sake of a cleaner notation,  $P/\sigma^2$  is taken as equal for uplink and downlink, but asymmetries could be had by simply discriminating the respective SNR variables. We set  $P/\sigma^2$  such that  $\text{SNR}_{n,k} = 20$  dB at a distance  $d$ , where  $d$  would be the inter-AP spacing if the network were arranged as a hexagonal grid. Under reasonable values for  $P$  and  $\sigma^2$ , this is compatible with a dense microcellular deployment [?].

In addition to  $G_{n,k}$ , the channel that connects the  $k$ th user with the  $n$ th AP includes a small-scale fading coefficient  $h_{n,k} \sim \mathcal{N}_{\mathbb{C}}(0, 1)$ , independent across users and APs. Each fading coefficient is presumed known by the respective receiver, an amply justified premise in cellular systems [?, ch. 4].

### B. Uplink

In the uplink, the  $n$ th AP is only interested in the signal from the  $n$ th user, the one in its cell, while the rest of users constitute interference. The observation at such  $n$ th AP is

$$y_n = \sqrt{G_{n,n}}h_{n,n}\sqrt{p_n P}s_n + \sum_{k \neq n} \sqrt{G_{n,k}}h_{n,k}\sqrt{p_k P}s_k + v_n, \quad (2)$$

where  $s_n$  is the unit-power symbol emitted by user  $n$  whereas  $p_n \in [0, 1]$  is its power control coefficient and  $v_n \sim \mathcal{N}_C(0, \sigma^2)$  is the noise. The SINR for the  $n$ th user is then

$$\text{sinr}_n^{\text{UL}} = \frac{p_n G_{n,n} |h_{n,n}|^2}{\sum_{k \neq n} p_k G_{n,k} |h_{n,k}|^2 + \sigma^2/P} \quad (3)$$

$$= \frac{p_n \text{SNR}_{n,n} |h_{n,n}|^2}{\sum_{k \neq n} p_k \text{SNR}_{n,k} |h_{n,k}|^2 + 1} \quad (4)$$

and its expectation over the small-scale fading yields the operationally meaningful local-average SINR [?], namely

$$\mathbb{E}[\text{sinr}_n^{\text{UL}}] = \mathbb{E} \left[ \frac{p_n \text{SNR}_{n,n} |h_{n,n}|^2}{\sum_{k \neq n} p_k \text{SNR}_{n,k} |h_{n,k}|^2 + 1} \right]. \quad (5)$$

In order to formulate loss functions that, as desired, drive the NN based only on large-scale quantities, it is convenient to have an SINR expression that depends on those quantities without the cumbersome outer expectation over the small-scale fading. To that end, replacing  $|h_{n,n}|^2$  and  $|h_{n,k}|^2$  by their expected value (unity), we obtain the very accurate proxy

$$\text{SINR}_n^{\text{UL}} = \frac{p_n \text{SNR}_{n,n}}{\sum_{k \neq n} p_k \text{SNR}_{n,k} + 1}. \quad (6)$$

### C. Downlink

In the downlink, the local-average SINR at user  $n$  is the dual of (??), precisely

$$\mathbb{E}[\text{sinr}_n^{\text{DL}}] = \mathbb{E} \left[ \frac{p_n \text{SNR}_{n,n} |h_{n,n}|^2}{\sum_{k \neq n} p_k \text{SNR}_{k,n} |h_{k,n}|^2 + 1} \right] \quad (7)$$

and its large-scale proxy is

$$\text{SINR}_n^{\text{DL}} = \frac{p_n \text{SNR}_{n,n}}{\sum_{k \neq n} p_k \text{SNR}_{k,n} + 1}. \quad (8)$$

## III. LOSS FUNCTIONS

Armed with suitable proxies for the local-average SINRs, loss functions of interest can now be formulated. First, we consider a max-min loss, attractive when the sole focus is fairness, and we suitably soften the function to allow relaxing that focus individually for each user. Then, we consider a max-product loss, enticing when the objective is to favor the average performance without a complete sacrifice in fairness.

### A. Soft Max-Min

The soft max-min function is

$$L_{\text{MM}} = \frac{1}{N} \underbrace{\left[ \sum_{n=0}^{N-1} \exp \left( \frac{\alpha_n}{(\text{SINR}_n + 0.01)^{0.4}} \right) \right]}_{\text{Objective}} + 0.1 \underbrace{\left[ \sum_{n=0}^{N-1} [p_n - 1]^+ \right]}_{\text{Constraints}} \quad (9)$$

where  $\{\alpha_n\}$  are regulating factors and  $[z]^+ = \max(0, z)$ . The cross-validation process described in [?] yields a multiplier of 0.1 for the constraints, relative to the objective. Readers

interested in the probability that some constraint is violated are referred to [?], where such probability and its impact on the ultimate performance is quantified for the more involved constraints that arise in centralized radio-access networks.

As  $\{\alpha_n\}$  grow large,  $L_{\text{MM}}$  becomes dominated by the lowest SINR and the optimization hardens to a maximization of such smallest SINR, i.e., to a max-min policy. Conversely, for decreasing  $\{\alpha_n\}$ , this max-min behavior softens as SINRs other than the lowest become progressively relevant.

The small offset 0.01 added to  $\text{SINR}_n$  avoids having the loss being dragged down by users below  $-20$  dB and prevents numerical problems in the NN learning stage. In turn, the exponent 0.4 compresses the dynamic range, improving the high-SNR performance and making the learning more stable.

### B. Max-Product

Our second loss function relates to the product of the SINRs, whose maximum is a desirable operating point in wireless systems [?, sec. 7.5]. The maximization of  $\prod_{n=0}^{N-1} \text{SINR}_n$ , or equivalently of its logarithm, amounts to the minimization of

$$L_{\text{MP}} = \frac{1}{N} \underbrace{\left[ \sum_{n=0}^{N-1} \beta_n \log \left( 0.01 + \frac{1}{\text{SINR}_n + 0.01} \right) \right]}_{\text{Objective}} + 0.1 \underbrace{\left[ \sum_{n=0}^{N-1} [p_n - 1]^+ \right]}_{\text{Constraints}} \quad (10)$$

where the multiplier applied to the constraints is again 0.1.

When the regulating factors  $\{\beta_n\}$  are equal, the combination  $\{\text{SINR}_n\}$  that minimizes  $L_{\text{MP}}$  exhibits satisfactory properties in terms of the tradeoff between average performance and fairness [?, sec. 7.5]. And, by tinkering with  $\{\beta_n\}$ , specific users could be afforded higher performance priorities.

Again, the offset shifting  $\text{SINR}_n$  by 0.01 avoids it being pulled down by users below  $-20$  dB while a second offset added to  $\frac{1}{\text{SINR}_n + 0.01}$  lessens the pull of users above 20 dB.

## IV. CENTRALIZED POWER CONTROL

Connecting back with the parametric optimization description in [?], the input parameter  $\theta$  consists of  $\{\text{SNR}_{n,k}\}$  while the optimization vector  $x$  packs the  $N$  power coefficients  $\{p_n\}$ , each constrained to the convex interval  $[0, 1]$ . The loss function is given by either  $L_{\text{MM}}$  or  $L_{\text{MP}}$ . And, as the transmit power in each cell is adjusted on the basis of the SNRs for all the users in the system, the power control is centralized.

### A. Learning Stage

The learning pipeline is illustrated in Fig. ?? while the parameters of the NN are summarized in Table ?. For preprocessing purposes,  $\{\text{SNR}_{n,k}\}$  are first converted to  $\{\log \text{SNR}_{n,k}\}$  and subsequently rendered zero-mean and unit variance. The processing then starts through an input layer equipped with rectified linear unit (RLU) activation functions. After feature extraction by this input layer, a hidden layer processes the data also via RLUs, and an output layer with

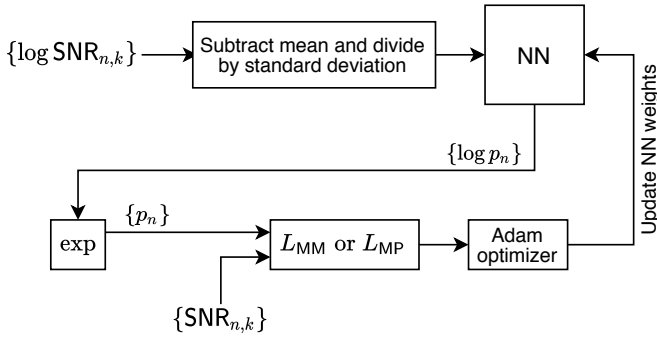


Fig. 1. Learning pipeline.

TABLE I  
NN SETTINGS.

	Input layer	Hidden layer	Output layer
Neurons	1200	200	$N = 30$
Activation function	RLU	RLU	Linear
Regularization	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$	L2 norm $\lambda = .001$

linear activation functions generates power coefficients in log scale; this guarantees positive outputs and averts numerical problems. From the SNRs and the corresponding NN outputs, the loss function of choice is quantified and an Adam optimizer is applied to minimize it. To avoid oscillations around local optima during the weight adjustment, the learning rate is reduced gradually from 0.001 down to 0.0001. And, to prevent overfitting, L2-norm regularization is employed in conjunction with the Adam optimizer: a portion  $\lambda = 0.001$  of the L2 norm of the weights is added to the loss.

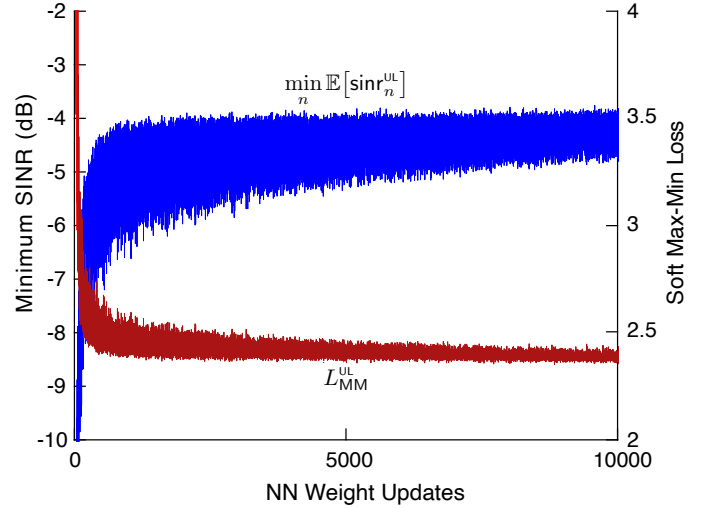
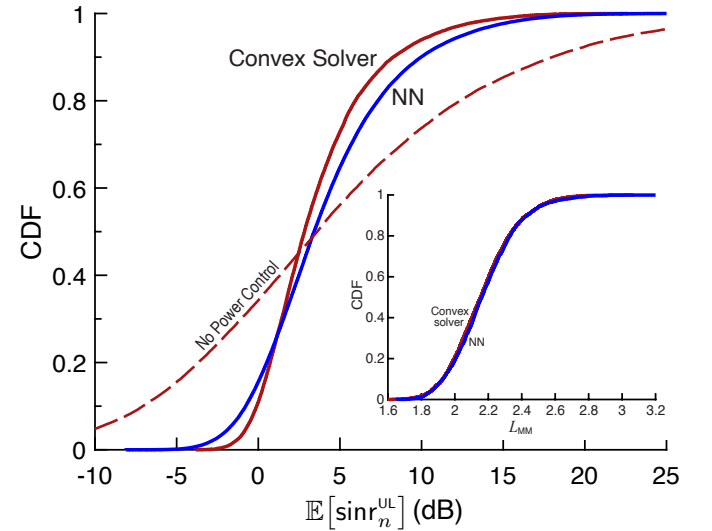
To streamline the learning, rather than a single large database, 1000 databases of 10000 system realizations are generated and, over each, 10 updates of the NN weights take place; each update relies on a randomly selected batch of 1000 realizations. The initialization is also random.

As an example, the uplink learning curve for the soft max-min loss function with  $\alpha_n = 1$  is provided in Fig. ??.

### B. Performance Evaluation

For both of the introduced loss functions, the cellular power control problem admits convex formulations in both uplink and downlink. Thus, performance benchmarks can be produced with an off-the-shelf convex solver [?].

We consider a system with  $N = 30$  cells, and equip the NN's input layer with 1200 neurons, somewhat above the minimum of  $N^2$ . The performance is evaluated by means of the cumulative distribution function (CDF) of local-average SINRs over the ensemble of possible user and AP positions. The NN and the convex solver are driven by the proxies in (??) and (??), for uplink and downlink respectively, while the performance with the obtained powers is assessed by means of the actual local-average SINRs in (??) and (??).

Fig. 2. Uplink soft max-min learning curve with  $\alpha_n = 1$ : loss and minimum local-average SINR as a function of the NN weight updates.Fig. 3. CDF of  $\mathbb{E}[\text{sinr}_n^{\text{UL}}]$  for the soft max-min loss function ( $\alpha_n = 1$ ): NN vs convex solver, alongside the baseline without power control ( $p_n = 1$ ). In the inset, CDF of  $L_{\text{MM}}$  for the NN and the convex solver.

Presented in Fig. ?? is the uplink performance for the soft max-min loss function with  $\alpha_n = 1$ . The NN yields transmit powers that, relative to the convex solver, favor high-SNR users slightly over their low-SNR brethren. In terms of the loss function, the NN is uniformly shy of the convex solver, but the agreement is extremely satisfying. The figure also depicts the baseline performance in the absence of power control, which confirms the importance of optimizing the transmit powers to avoid enormous disparities among users. For the max-product loss function, Fig. ?? reveals an even better match between the NN and the convex solver. The CDFs of local-average SINRs again cross over while, in terms of the loss function, the NN is uniformly shy of the convex solver.

In the downlink (see Figs. ?? and ??), the importance of power control is somewhat toned down because the intercell interference distribution is more benign [?]. Indeed, a contrast

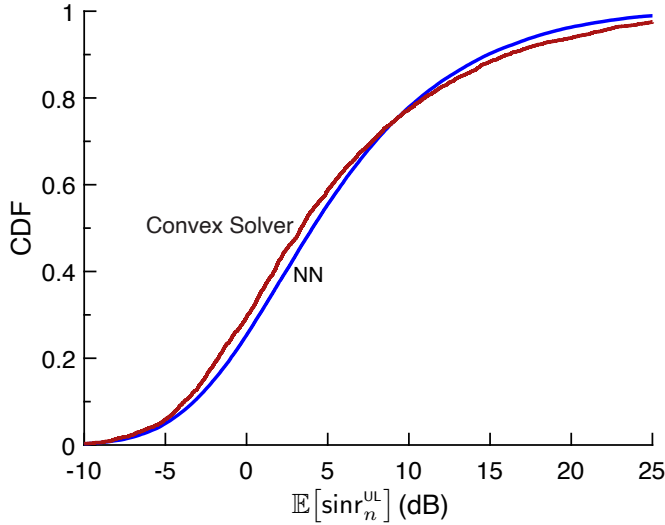


Fig. 4. CDF of  $\mathbb{E}[\text{sinr}_n^{\text{UL}}]$  for the max-product loss function ( $\beta_n = 1$ ): NN vs convex solver.

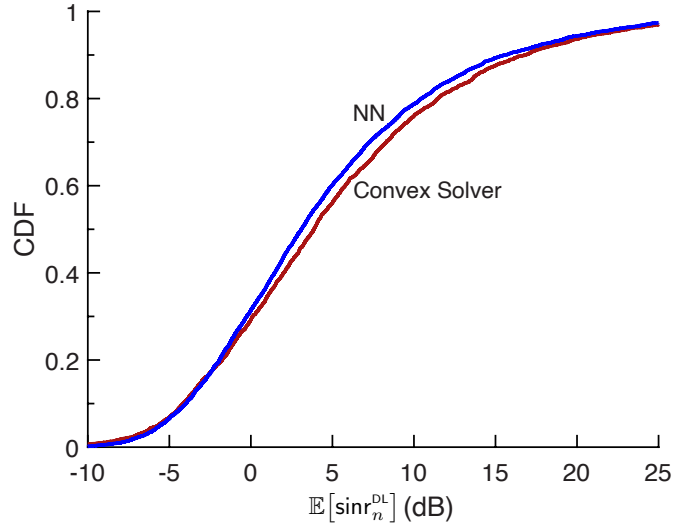


Fig. 6. CDF of  $\mathbb{E}[\text{sinr}_n^{\text{DL}}]$  for the max-product loss function ( $\beta_n = 1$ ): NN vs convex solver.

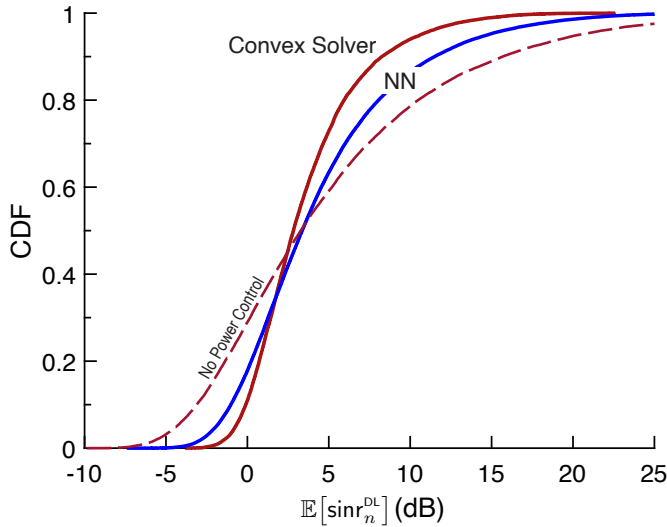


Fig. 5. CDF of  $\mathbb{E}[\text{sinr}_n^{\text{DL}}]$  for the soft max-min loss function ( $\alpha_n = 1$ ): NN vs convex solver. Also shown is the baseline performance without power control, i.e., with  $p_n = 1$ .

of Figs. ?? and ?? evidences that a non-power-controlled transmission approximates fairly closely the max-product solution.

### C. Computational Cost

As a measure of the computational cost, we invoke the running time on a common computational platform. And, to err on the side of caution, a CPU-based platform befitting the convex solver is chosen.

Each NN inference is in essence a matrix multiplication, over two orders of magnitude faster than a convex solver for our 30-cell system. For a growing number of cells, this advantage is likely to increase further.

In terms of learning, which needs to take place upon changes in the system or the environment, it is interesting to contrast the learning time of our unsupervised NN with that of a supervised NN of the same dimensions. While the training

effort per parameter sample is essentially the same in both cases, a supervised NN would require producing those samples by solving the corresponding optimization problems in the first place, again resulting in the over-two-order-of-magnitude advantage for the example at hand. This primacy could be tempered slightly by the possibility of reusing training samples on the part of the supervised scheme, allowing it to cope with a fraction of the data, but in a worst-case scenario the unsupervised approach would retain a net advantage of over one order of magnitude—which would then rise with the number of cells. (Allowing the NN to train on a more apt GPU-based platform, this superiority would increase by yet another order of magnitude.)

## V. DISTRIBUTED POWER CONTROL

While centralizing the power control is advantageous from a performance standpoint, allowing it to operate distributedly—each cell on its own—is preferable from an implementation perspective. We can shift towards distributedness by restricting the input parameter  $\theta$  to contain, rather than  $\{\text{SNR}_{n,k}\}$ , only  $\{\text{SNR}_{n,n}\}$ . The transmit powers are then optimized based solely on same-cell SNRs, and owing to the reduced parameter space, the training is considerably more expeditious.

For distributed power control, an additional benchmark exists, namely the so-called fractional power control policy [?]. This is in fact the mainstream approach to power control in current cellular networks, with the remarkably simple form

$$p_n \propto \frac{1}{\text{SNR}_{n,n}^{\vartheta}}, \quad (11)$$

where  $\vartheta$  determines the extent to which fairness is promoted [?]; typical values in 4G and 5G systems are  $\vartheta \approx 0.5\text{--}0.7$ , with lower values favoring the average performance while higher values foster fairness and cell-edge robustness. The proportionality factor in (??) should be such that  $p_n \in [0, 1]$ .

Presented in Fig. ?? is the comparison between the NN-based approach (with soft max-min and  $\alpha_n = 1$ ) and fractional power control (with  $\vartheta = 0.7$ ). It can be surmised that:

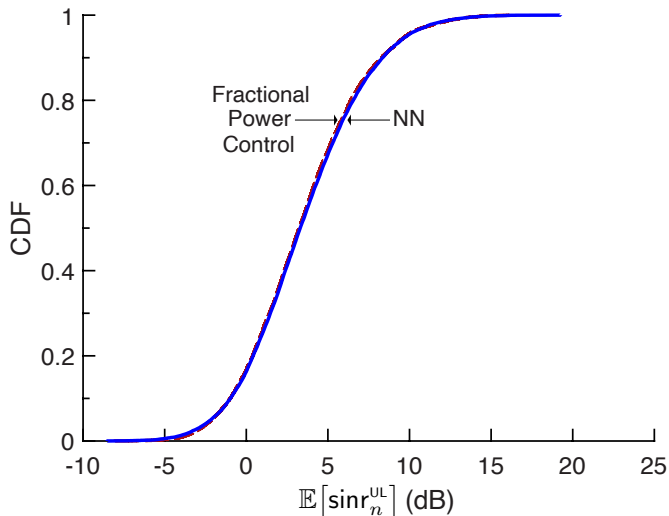


Fig. 7. CDF of  $\mathbb{E}[\text{sinr}_n^{\text{UL}}]$ : NN for soft max-min ( $\alpha_n = 1$ ) in solid versus fractional power control ( $\vartheta = 0.7$ ) in dashed.

- Distributed power control is only modestly worse than centralized power control (cf. Figs. ?? and ??).
- Fractional power control is indeed extremely effective, its performance almost identical to that of the NN—despite the fact that the NN has access to the same-cell SNRs for the entire system while fractional power control truly operates on a cell-by-cell basis.

This latter point invites a dual remark. On the one hand, the NN is useful to certify the effectiveness of fractional power control and to calibrate the equivalence of the parameter  $\vartheta$  in terms of any well-defined loss function. On the other hand, a solution far simpler and equally performing to the NN is seen to exist—less general, but emanating from the core structure of the problem at hand. Precisely, fractional power control was derived analytically as the strategy that minimizes the variance of the SIR (in dB) in a two-cell setup [?].

## VI. SUMMARY

The unsupervised-learning approach to parametric optimization described in [?] is very effective for cellular power control, adding to the body of works that leverage NNs to tackle wireless communication problems [?], [?], [?], [?], [?], [?], [?], [?], [?].

With a view to having suitable benchmarks, only objectives leading to convex formulations have been considered. Other objectives can be defined that do not admit convex forms, in particular the maximization of weighed sum of the spectral efficiencies. Although benchmarks may then be lacking, these objectives can be tackled all the same with our approach, and instances of nonconvex objectives are given in [?] for centralized radio-access networks.

The distributed uplink setting dispenses a reminder of the fact that, whenever sound models are available that describe the structure of the problem, solutions that are simpler and even illuminating may sometimes be derived directly from those models. Learning-based approaches should complement and expand, rather than replace, such model-based derivations.

The Python code developed to produce the results in this letter is openly available [?].

## ACKNOWLEDGMENT

The invitation from the Editor-in-Chief, the handling of the Associated Editor, and the constructive feedback from the anonymous reviewers, as well as from Dr. Giovanni Geraci, are all gratefully acknowledged.