International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 8 (2019), pp. 1200-1204

Tabu Search Based Pilot Allocation Algorithm in Massive MIMO

Rand Abdul Hussain[1] and Josko Zec[2]

[1]*Ph.D., Student, Department of Computer Engineering and Sciences, Florida Institute of Technology, USA.*
ORCID of Author 1: 0000-0001-9596-3050

[2]*Associate Professor, Department of Computer Engineering and Sciences, Florida Institute of Technology, USA.*
ORCID of Author 2: 0000-0003-1649-1222

Abstract

In this paper, we have proposed a tabu search (TS) based algorithm to optimize the allocation of pilots to the users during the uplink (UL) pilot transmission phase to alleviate the pilot contamination problem. The goal is to maximize the minimum UL signal-to-interference ratio (SIR) in the asymptotic massive multiple-input multiple-output (MIMO) system. The pilot allocation problem is combinatorial, and it is generally non-deterministic polynomial-time hard (NP-hard). We have proposed an algorithm that converges to a near optimal solution. Performance evaluation of our algorithm has been carried against that of exhaustive search and state-of-the-art algorithm where the latter can be applied when the search space is prohibitively large for an exhaustive search. Simulation results have confirmed the superiority of our algorithm over the related published work.

Keywords: Pilot contamination, Pilot assignment, NP-hard.

I. INTRODUCTION

Massive multiple-input multiple-output (MIMO) is a cellular network technology that has been proposed in [1] to increase the capacity of the network by adding more antennas to cell sites. Specifically, a very large number of antennas are deployed at a base station (BS), and it typically serves many single-antenna user equipment (UEs) simultaneously [2]. As a BS needs to estimate the channels from its serving UEs [3], each UE is assigned a unique pilot sequence where the set of these sequences are mutually orthogonal. Since the channel responses are static during time/frequency blocks, which is called coherence channel block [4], coherence block consists of a limited number of samples. Some of these samples, in a typical time-division duplex (TDD) massive MIMO system, are reserved for the uplink pilot signals while the remaining samples are for the uplink and downlink data transmission. Hence, pilots are overhead, and it is efficient to reserve less than half the samples in the coherence block to transmit them [4]. Therefore, the pilot sequence length is limited, and the number of unique mutual orthogonal pilots is limited. Hence, reusing the pilots by UEs across the cells is in inevitable. As a result, the asymptotic analysis in [1] has shown that as the number of antennas at the BS reaches infinity, the uncorrelated thermal noise, intra-cell interference, and fast fading effects vanish, and the only impairment resulted from the UEs that use the same pilot as the served user remains in the effective signal-to-interference ratio (SIR) calculated for that UE. This is referred to as *pilot contamination*.

II. RELATED WORK

Various solutions have been proposed to mitigate the effect of the pilot contamination by optimizing the pilot allocation to the UEs. The work in [5] has designed a greedy tabu search (GTS) algorithm with exponential complexity. Therefore, GTS is only applicable when the number of cells and users is small. The approach presented in [5] has adopted the tabu search (TS) method from [6], which has polynomial complexity and performance below the GTS. The goal of both algorithms is to maximize the normalized system data rate expressed in bits/second/Hertz. The authors in [7] have proposed a deep learning-based pilot allocation scheme (DL-PAS) to reduce the computational time related to pilots' assignment algorithms. DL-PAS is a deep multilayer perceptron system where the training data is the optimal pilot assignments (acquired through exhaustive search) with UEs' locations. Then, DL-PAS provides the pilot assignment given UEs' locations by inferring the relationship between the pilot allocation and the UEs' locations. However, the computational complexity in DL-PAS is in the training phase because the size of the output label is factorial to the number of UEs in a cell, which is still not practical.

The goal of the scheme in [8] is to maximize the minimum achievable rate for the target cell by assigning pilots used in the target cell to the UEs in the other cells in a way that minimizes the inter-cell interference to the UEs in the target cell. The authors in [9] have proposed an iterative pilot allocation scheme where each cell solves its optimization problem of maximizing the minimum UL signal-to-interference-plus-noise-ratio (SINR) of its UEs. In the smart pilot assignment algorithm (SPA) [9], a cell assigns the UE with worst channel quality a pilot sequence that causes the least inter-cell interference, and it is carried out for all the UEs in the cell. Then, the algorithm is applied sequentially to the other cells in the system, and the procedure is iteratively applied to maximize the minimum UL SINR of all the UEs. Therefore, SPA always accepts the solution proposed by each cell. While the schemes in [5] are also based on TS, the proposed algorithm is different in the fitness function, tabu list, and the mechanism of exploring the neighborhood. E.g., the proposed algorithm depends on two matrices to set the forbidden rules and the stopping rule vs. a list with a specific length [5] that records the solution vectors of past few iterations (operates like a queue) to prohibit cycling. Like [9], our goal is to maximize the minimum UL SINR of all the UEs in the system. Thus, we started by regenerating [9] to serve as baseline evaluating our results when the exhaustive search is not feasible.

III. SYSTEM MODEL

We consider the UL of time-division duplex (TDD) multi-cell massive MIMO system. The system consists of L hexagon cells, where each cell has one BS in the center. The BS is deployed with M antennas and serves K single-antenna UEs [1]. The channel vector from the k-th user in the j-th cell to the l-th BS is denoted by $\mathbf{h}_{jk}^{l} \in \mathbb{C}^{M}$. And $\mathbf{h}_{jk}^{l} = \mathbf{g}_{jk}^{l} (\beta_{jk}^{l})^{1/2}$ where $\mathbf{g}_{jk}^{l} \sim CN(\mathbf{0}, \mathbf{I}_M)$ denotes the small-scale fading vector and $\beta_{jk}^{l} = z_{jk}^{l} / (r_{jk}^{l})^{\alpha}$ denotes the large-scale fading coefficient where z_{jk}^{l} denotes the shadow fading, and $10\log_{10}(z_{jk}^{l})$ is distributed as Gaussian with zero mean and standard deviation of σ_{dB}^{shad}; r_{jk}^{l} is the distance between the k-th user in the cell j to the l-th cell, and α is the path loss exponent.

During the UL pilot transmission, a pilot is assigned to each UE so that the BS can estimate the channel responses from its intended UEs during the current coherence block [3]. Pilot sequences are designed to be mutually orthogonal to get rid of the interference caused by the transmitting UEs other than the intended UE. Since the channel coherence time limits the number of unique pilots that can be assigned to the UEs in the system [3], typically, the UEs in the same cell are assigned unique pilots (no intra-cell interference), while the same set of pilots are reused in every cell. Therefore, using the maximum ratio combining (MRC), during the UL data transmission phase, the UL SINR of the k-th user in the j-th cell can be calculated as [9]:

$$SINR_{jk} = \frac{|(\mathbf{h}_{jk}^{j})^{H}\mathbf{h}_{jk}^{j}|^2}{\Sigma_{l\neq j} |(\mathbf{h}_{lk}^{j})^{H}\mathbf{h}_{lk}^{j}|^2 + |v_{jk}|^2/\rho_d} \tag{1}$$

where v_{jk} denotes the uncorrelated noise and the intra-cell interference, and ρ_d denotes the power of the transmitted UL data. As $M \to \infty$, only inter-cell interference that results from reusing the same pilot sequence remains, and (1) converges to below [1]:

$$\gamma_{jk} = \frac{(\beta_{jk}^{j})^2}{\Sigma_{l\neq j} (\beta_{lk}^{j})^2} \tag{2}$$

where γ_{jk} is the effective UL SIR, and it only depends on the large-scale fading coefficients. Similar to the related works [5], [7-9] our proposed algorithm exploits the large-scale fading coefficients to optimize the pilot allocation, and our goal is to maximize the minimum uplink SIR in the system.

IV. PROPOSED ALGORITHM

The proposed TS based algorithm starts with the initial UL SIRs of the UEs in the system that are obtained by random pilot allocation, and the SIRs are calculated using (2). Hence, when each cell serves K UEs, each pilot will be shared by L UEs. Denote $\mathbf{P}$, a pilot allocation $K\times L$ matrix with each entry U_{li}^{k} represents an i-th UE from the l-th cell allocated the k-th pilot, and $i = \{1, ..., K\}$.

Hence, the k-th row vector of $\mathbf{P}$, $\mathbf{p}_k \triangleq \mathbf{P}(k, :) = \{U_{1i}^{k}, \dots, U_{Lx}^{k}\}$ is a group of UEs that share the same pilot, and $x = \{1, ..., K\}$. Accordingly, the l-th column of $\mathbf{P}$ represents the UEs that are served by the l-th cell that each is assigned a unique pilot. Similarly, denote $\mathbf{S}$ a $K\times L$ matrix that stores the SIRs of the UEs that share pilots as recorded in the $\mathbf{P}$ matrix. Hence, $\mathbf{s}_k \triangleq \mathbf{S}(k, :) = \{\gamma_{U_{1i}^{k}}, \dots, \gamma_{U_{Lx}^{k}}\}$ is the SIRs of the UEs that are assigned the k-th pilot. Since the goal is to maximize the minimum UL SIR in the network, the algorithm computes the minimum SIR among the UEs that share the same pilot as below:

$$R(\mathbf{s}_k) = \min(\gamma_{U_{1i}^{k}}, \dots, \gamma_{U_{Lx}^{k}}) \tag{3}$$

where $R(\mathbf{s}_k)$ is the minimum SIR among the group of UEs of the k-th pilot ($\mathbf{p}_k$). In this work, the *tabu list* is a $K\times KL$ matrix where the k-th row represents the forbidden users for the k-th pilot, and the tabu list, which is denoted by $\mathbf{T} = [t_{<k, U_{li}^{k}>}]_{K\times KL}$ will not erase any forbidden UEs until the algorithm stops. We also define the matrix $\mathbf{D} = [d_{<k, l>}]_{K\times L}$ to record the cells that will not change pilots' allocation to their UEs to enhance the $R(\mathbf{s}_k)$ of $\mathbf{p}_k$. Therefore, the $\mathbf{D}$ matrix is another memory structure for the tabu list. Also, the $\mathbf{D}$ matrix will lead the algorithm to convergence, and the algorithm will stop when all the entries of the $\mathbf{D}$ matrix change values to 1, which serves as the *stopping rule*.

In each iteration, the algorithm's objective is to raise the minimum SIR in the system. Hence, it selects the $\mathbf{p}$ vector with least $R(\mathbf{s})$ (*target* $\mathbf{p}$) and a second $\mathbf{p}$ vector with next least $R(\mathbf{s})$ to exchange UEs between these two vectors. Specifically, the algorithm swaps the UE with least SIR (least $R(\mathbf{s}_i)$) with UE of the same cell that belongs to the pilot vector of the next least $R(\mathbf{s})$. The change will be accepted if the minimum SIR among the UEs of the swapped vectors is greater than the least $R(\mathbf{s})$. If the change is accepted, the algorithm begins a new iteration. Otherwise, the tabu list $\mathbf{T}$ records the forbidden UE(s) (for the pilot vector(s)) that result(s) in lowering $R(\mathbf{s})$ less than the least SIR, and the algorithm selects another $\mathbf{p}$ vector with next least $R(\mathbf{s})$ for swapping with target pilot vector. If all (K-1) pilots' vectors have already been selected and the change has never been accepted, then the tabu list $\mathbf{D}$ records the forbidden cell index where the cells' UEs will not change their pilots to enhance the $R(\mathbf{s})$ of the target pilot.

Therefore, in each iteration, the neighborhood of the target $\mathbf{p}$ selected UE is the (K-1) UEs of the other (K-*1*) $\mathbf{p}$ vectors (one UE from each other pilot vector) that will be exchanged with selected UE. However, the algorithm does not explore all the neighborhood members and then selects the change to keep. Instead, the algorithm stops exploring the neighborhood once a change is accepted by the fitness function (unlike [5] where the neighborhood is explored before the acceptance decision). Hence, the *fitness function* is maximizing the minimum $R(\mathbf{s})$ among the $\mathbf{p}$ vectors. When the selected UE of the target cell explores all the members in the neighborhood, and the $R(\mathbf{s})$ of target pilot does not change, a new iteration begins with new UE that will be selected from the target $\mathbf{p}$ vector. The selected UE will be the one with the next lowest SIR in the target $\mathbf{p}$ vector. If all the UEs in the target $\mathbf{p}$ explore their neighborhood, which also means the tabu list $\mathbf{D}$ entries related to the target $\mathbf{p}$ vector all change to 1, then the algorithm takes the next pilot vector with least $R(\mathbf{s})$ as the target vector and the old target pilot $R(\mathbf{s})$ will be given the *Num* value. Therefore, the *Num* value is larger than any possible SIR, and the *Num* is increased and assigned to the target $\mathbf{p}$ pilot $R(\mathbf{s})$ when its UEs explore all their options. Still, the vector with *Num* value will be an option to exchange for other pilot vectors, and the true $R(\mathbf{s})$ (rather than the *Num* value) will be evaluated to decide on acceptance or rejection of the swap. In other words, the *aspiration criterion* is implemented through the tabu list $\mathbf{D}$ matrix. Although the forbidden pilot vector will lose the priority to explore, it is still an option for other pilots and improving its $R(\mathbf{s})$ is possible until the last $\mathbf{p}$ vector explores its options. The algorithm would accept the change to raise the $R(\mathbf{s})$ of this vector rather than the target vector if it would increase the minimum SIR in the system.

Below is the detailed pseudocode of the proposed algorithm:

Proposed Algorithm

Input: K, L, $\mathbf{P}$, $\mathbf{S}$, β_{lk}^{j}, $j = \{1, ..., L\}$, $l = \{1, ..., L\}$, $k = \{1, ..., K\}$, *Num*

Output: **P, S**

- $\mathbf{D} = \mathbf{0}$, $\mathbf{T} = \mathbf{0}$, $i_{\text{old}} = 0$, $c_{\text{old}} = 0$
- Calculate $R(\mathbf{s}_1), ..., R(\mathbf{s}_K)$ for $\mathbf{p}_1, ..., \mathbf{p}_K$ using (3)
- **while** $\exists\, d_{<k,\, l>} = 0$ **do**
  - Sort($\{R(\mathbf{s}_1), ..., R(\mathbf{s}_K)\}$) ascendingly and save the sorted values in $\mathbf{V}$ (1, :) and corresponding pilot number in $\mathbf{V}$ (2, :), where $\mathbf{V}$ is $2{\times}K$ matrix
  - $i = v_{<2,1>}$; $x = v_{<2,2>}$
    % $v_{<2,1>}$ and $v_{<2,2>}$ are entries in $\mathbf{V}$ matrix
  - Sort($\mathbf{s}_i$) ascendingly, and store the corresponding indices of the sorted values (cells' indices) in $\mathbf{o}$
  - $c = \mathbf{o}(1)$
  - **if** $((i = i_{\text{old}}) \wedge (c = c_{\text{old}})) \vee (d_{<i,\, c>} = 1)$
    - $count = 2$
    - $c = \mathbf{o}(count)$
    - **while** $(d_{<i,\, c>} = 1) \wedge (count \leq L)$
      - $count = count + 1$
      - $c = \mathbf{o}(count)$
    - **end while**
  - **end if**
  - $SIR_{\text{least}} = R(\mathbf{s}_i)$; $SIR_{\text{leastnew}} = \text{-inf}$; $count = 1$
  - **while** $(SIR_{\text{leastnew}} < SIR_{\text{least}}) \wedge (count < K)$
    - **if** $(t_{<i,\, \mathbf{p}x\,(c)>} = 1) \vee (t_{<x,\, \mathbf{p}i\,(c)>} = 1)$
      - **if** $count < (K\text{-}1)$
        - $count = count + 1$
        - $x = v_{<2,\, count>}$
        - **if** $count = (K\text{-}1)$
          - $d_{<i,\, c>} = 1$
        - **end if**
      - **else**
        - $c_{\text{old}} = c$; $i_{\text{old}} = i$; $d_{<i,\, c>} = 1$; $q = \sum_{l=1}^{L} d_{<i,\, l>}$
        - **if** $q = L$
          - Assign *Num* to the $R(\mathbf{s}_i)$; $Num = Num + 1$
        - **end if**
        - $count = K$
      - **end if**
      - continue
    - **end if**
    - $\mathbf{a_1} = \mathbf{p}_i$; $\mathbf{a_2} = \mathbf{p}_x$; Swap $\mathbf{a_1}(c)$ with $\mathbf{a_2}(c)$
    - Calculate SIRs for UEs in $\mathbf{a_1}$ using (2) and store in $\mathbf{b_1}$
    - Calculate SIRs for UEs in $\mathbf{a_2}$ using (2) and store in $\mathbf{b_2}$
    - Calculate $R(\mathbf{s}_x)$, $R(\mathbf{b_1})$, $R(\mathbf{b_2})$ using (3)
    - $SIR_{\text{least}} = \min(R(\mathbf{s}_{\text{i}}), R(\mathbf{s}_{\text{x}}))$
    - $SIR_{\text{leastnew}} = \min(R(\mathbf{b_1}), R(\mathbf{b_2}))$
    - **if** $SIR_{\text{leastnew}} > SIR_{\text{least}}$
      - Update $\mathbf{V}$ with values resulted from Swap
      - $SIR_{\text{least}} = \min(\mathbf{V}(1, :))$; $c_{\text{old}} = c$; $i_{\text{old}} = i$
      - **if** $SIR_{\text{least}} = SIR_{\text{leastnew}}$
        - **if** $R(\mathbf{b_1}) = SIR_{\text{leastnew}}$
          - $t_{<i,\, \mathbf{a2}(c)>} = 1$
        - **else if** $R(\mathbf{b_2}) > R(\mathbf{s}_x)$
          - $t_{<x,\, \mathbf{a1}(c)>} = 1$
          - **end if**
        - **end if**
      - **end if**
      - Update $\mathbf{P}$, $\mathbf{S}$, $R(\mathbf{s}_i)$, $R(\mathbf{s}_x)$
    - **else**
      - if $R(\mathbf{b_1}) = SIR_{\text{leastnew}}$
        - $t_{<i,\, \mathbf{a1}(c)>} = 1$
        - **if** $R(\mathbf{b_2}) < SIR_{\text{least}}$
          - $t_{<x,\, \mathbf{a2}(c)>} = 1$
        - **end if**
      - **else**
        - $t_{<x,\, \mathbf{a2}(c)>} = 1$
        - **if** $R(\mathbf{b_1}) < SIR_{\text{least}}$
          - $t_{<i,\, \mathbf{a1}(c)>} = 1$
        - **end if**
      - **end if**
    - **end if**
  - **end while**
- **end while**

V. SIMULATION RESULTS

In this section, we confirm the effectiveness of the proposed algorithm through Monte-Carlo simulations. The algorithm performance is compared with three alternatives. First, the trivial baseline of random pilot assignment (randomly assigns pilots to the users in each cell). Second, exhaustive search (evaluates $(K!)^{L\text{-}1}$ pilot combinations and choose the one that maximizes the minimum SIRs), and third, SPA algorithm (the best solution during the non-exhaustive set of iterations is chosen).

The system parameters used in all simulation are [1]: L cells with a radius of 1600m, the cell radius hole (no UE is served within this distance) is 100m, α (path loss exponent) =3.8, and σ_{dB}^{shad} (the shadow fading standard deviation) = 8dB. Each cell serves K UEs, where the UEs are uniformly distributed. Shadow fading is added such that a UEs is served by the BS, which provides it with the largest large-scale fading coefficient (β_{jk}^{j}) [3].

Fig. 1 depicts the cumulative distribution function (CDF) of the minimum UL SIR in the system when L=3 and K=6. For this small system, exhaustive search is feasible and can be used as the performance benchmark limit. It is evident that our proposed algorithm's performance surpasses that of SPA and the conventional method. For example, the probability of the minimum uplink SIR (dB) below 5 dB is approximately 5% using the exhaustive search, 12% using our proposed algorithm (iterated KLL times), 25% using SPA (with KLL iterations), 35% using SPA (with KL iterations), 49% using SPA (with L iterations), and 86% using the conventional approach. The only method outperforming our proposed algorithm is the exhaustive search.

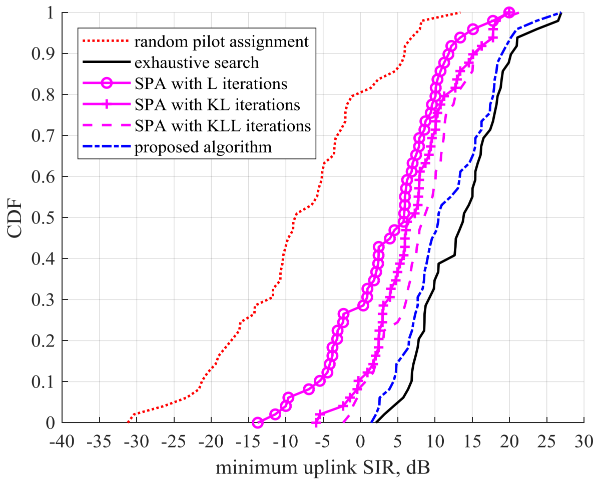

Fig. 1 CDF of the minimum UL SIR [dB] among all the UEs.

Fig. 1 also shows the more SPA iterated, the better its performance; however, letting SPA iterated the same number as the proposed algorithm, the proposed algorithm outperforms SPA. Moreover, SPA computational complexity in each iteration is (LK log K) [9], while the proposed algorithm computational complexity is (K log K + L log L) where K log K results from sorting the R(**s**) related to pilot vectors, and L log L results from sorting the SIRs of UEs of the pilot with least R(**s**). Hence, the computational complexity of the SPA is more than the proposed algorithm. Also, the SPA as the proposed algorithm calculates SIRs of the UEs using (2) (L^2 is the cost of computing the SIRs for two pilots' UEs). Then, the total complexity of the proposed algorithm is O (N max {K log K, KL^2}), where N is the number of iterations and $N = KL^2$.

Fig. 2 shows the convergence of the minimum UL SIR in the network (same setting as Fig. 1) where the minimum UL SIR is recorded in each iteration for all the realizations. Then, the mean of the realizations has been taken for each iteration. However, since SPA has L values of minimum SIR in each iteration, we first took the maximum SIR among the minimum SIRs of the L iterations, while in the second method we took the mean of the minimum SIRs in the L iterations. Then for both methods, we took the mean of the realizations for each iteration. It is obvious that the proposed algorithm converges around 30 iterations (when the **D** matrix entries change to 1) while the SPA algorithm has no convergence pattern.

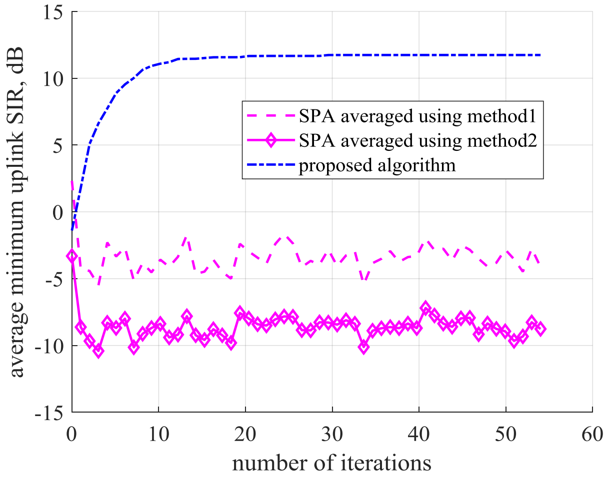

Fig. 2 The convergence of the average minimum uplink SIR (dB) in the system.

Finally, Fig. 3 shows the CDF of the UL SIRs in the network when K=40 (is typical in massive MIMO systems [2,3]) and L=19 (two hexagonal tiers). The horizontal line indicates the 4 percent value, and the vertical line indicates 10 dB. At the 96%-likely UL SIR point, UL SIR is around 10dB using the proposed pilot allocation algorithm (iterated KLL times) while it is about -6dB using the SPA algorithm (iterated KLL times), and -8dB using the conventional approach.

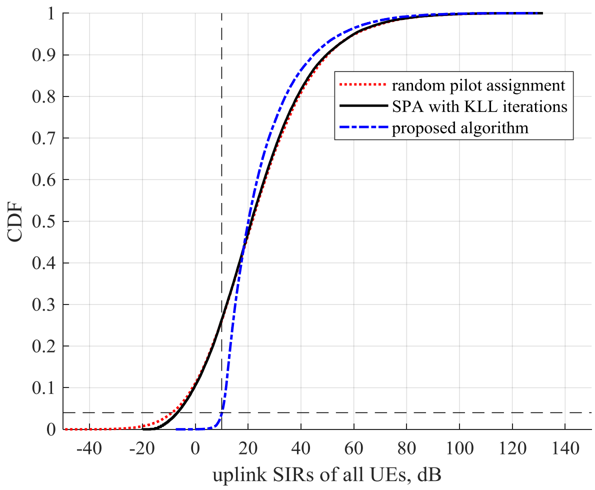

Fig. 3 CDF of the UL SIRs [dB] for all the UEs. The horizontal and vertical lines represent 0.04 value and 10dB, respectively.

An additional merit of the proposed algorithm over state-of-the-art algorithm (SPA) is the mechanism of learning. Although the limitation of the coherence time requires immediate calculations to the SIRs using the large-scale coefficients instead of learning from the sensed values (SIRs), the proposed algorithm could learn from the environment using the tabu list. In contrary, the SPA needs to prioritize the assignment task according to the precise information about, for example, the large-scale characteristics of the fading channels between UEs and their serving BS to assign pilots to the cell's UEs.

VI. CONCLUSION

We have proposed an algorithm that optimizes pilot assignment to the UEs in the asymptotic massive MIMO system. The proposed algorithm is superior to the state-of-the-art in terms of computational complexity, accuracy, and it has an inherent learning mechanism.

REFERENCES

[1] Marzetta, T.L., 2010: "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas", IEEE Trans. Wirel. Commun., 9 (11), pp. 3590–3600.

[2] Chien, T.V. and Björnson, E., 2017: "Massive MIMO Communications" in Xiang, W., Zheng, K., Shen, X. (Ed.): '5G Mobile Communications' (Springer, Cham).

[3] Björnson, E., Hoydis, J. and Sanguinetti, L., 2017: "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency", Foundations and Trends® in Signal Processing, 11(3-4), pp. 154–655.

[4] Björnson, E., Larsson, E.G. and Debbah, M., 2016: "Massive MIMO for Maximal Spectral Efficiency: How Many Users and Pilots Should be Allocated?", IEEE Trans. Wirel. Commun., 15(2), pp. 1293–1308.

[5] Jin, S., Li, M., Huang Y., Du, Y., Gao, X., 2015: "Pilot Scheduling Schemes for Multi-cell Massive Multiple-Input-Multiple-Output Transmission", IET Commun., 9(5), pp. 689-700.

[6] Srinidhi, N., Datta, T., Chockalingam, A. and Rajan, B.S., 2011: "Layered Tabu Search Algorithm for Large-MIMO Detection and a Lower Bound on ML Performance", IEEE Trans. Wirel. Commun., 59(11), pp. 2955–2963.

[7] Kim, K., Lee, J. and Choi, J., 2018: "Deep Learning Based Pilot Allocation Scheme (DL-PAS) for 5G massive MIMO System ", IEEE Commun. Lett., 22(4), pp. 828-831.

[8] Alkhaled, M., Alsusa, E. and Hamdi, K. A., 2017: "A New Allocation Algorithm for Pilot Contamination Mitigation in TDD Massive MIMO Systems", Proc. 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA.

[9] Zhu, X., Wang, Z., Dai, L., Qian, C., 2015: "Smart Pilot Assignment for Massive MIMO", IEEE Commun. Lett., 19(9), pp. 1644-1647.