Fast Optimal Antenna Selection for Massive MIMO

Ke He, Thang X. Vu, Symeon Chatzinotas, and Björn Ottersten

The Interdisciplinary Centre for Security, Reliability and Trust (SnT) - University of Luxembourg,
L-1855 Luxembourg. E-mail: {ke.he, thang.vu, symeon.chatzinotas, bjorn.ottersten}@uni.lu

***Abstract*—This paper investigates the optimal antenna selection problem for very large-scale multiple-input multiple-output (MIMO) systems. The problem is NP-hard, and solving it optimally requires exponential computation time, which limits its application in very large MIMO systems. In order to develop a feasible optimal method and help evaluate the performance gap of low-complexity algorithms in the scenarios of our interests, we start by proposing a new objective function, which reformulates the original problem into a shortest path finding. By combining the best-first search strategy, we further develop an efficient pruning algorithm, namely BFS-AS, for finding the optimal antenna combination. Our simulation results show that the proposed BFS-AS not only achieves the exact optimal performance, but also has a fixed space complexity and a much lower average time complexity with comparison to the existing optimal performance achieving approaches.**

***Index Terms*—Antenna selection, massive MIMO, best-first search, branch-and-bound.**

I. Introduction

Massive multiple-input multiple-out (MIMO) has been widely deployed to support lower latency, high capacity, and massive connectivity in wireless communications [1]. Meanwhile, the use of a large antenna array not only boosts the system spectral efficiency, but also brings diverse challenges in the applications of massive MIMO. One important issue of such challenges would be the tremendous difficulty on optimal antenna selection in very large-scale systems [2]–[4].

In practical scenarios, a reduced number of radio frequency (RF) chains, compared to the large number of antennas, will be deployed at the base station (BS) with economical consideration [1], [5]. This limitation imposes the concept of antenna selection in massive MIMO, which aims to achieve higher channel capacity with reduced hardware and operational costs [5]. This objective can be accomplished by switching RF chains to the best subset of antennas. Existing literature has shown that antenna selection can achieve effective higher system performance than that of the same system without selection [4]. However, finding the optimal antenna combination requires a computational complexity growing exponentially with the increasing problem scale, which is obviously challenging for massive MIMO.

With the purpose of achieving a good system performance and meanwhile reducing the computational complexity, various low-complexity sub-optimal antenna selection schemes have been proposed during the last decade. In [2], the author proposed a norm-based sub-optimal selection algorithm by iteratively removing the antenna having least contribution to the channel capacity. The proposed norm-based selection yields reduced complexity. In addition, the authors proposed a greedy search scheme in [3]. Instead of removing one antenna at each iteration, the greedy selection appends antennas which contribute the most to the channel capacity during iterations. Since practical systems typically have much lower number of RF chains than the equipped antennas, greedy search is more computationally efficient than the norm-based algorithm. Although the simulation results showed that greedy search is near-optimal in small antenna-array configurations, it still involves considerable performance loss in very large-scale systems [6].

In recent years, with the wide application of machine learning [7], the authors of [8] proposed an intelligent Monte-Carlo tree search algorithm for the selection of massive antennas. The authors therein showed that the Monte-Carlo tree search algorithm achieves better performance than the greedy selection. For the practical scenarios, the authors of [9] proposed a learning based antenna selection to improve the system efficiency with imperfect channel state information (CSI). Moreover, the authors of [10] overcame the non-convexity by proposing a joint antenna selection and precoding design, which maximizes the system performance to practical constraints. However, the performance gaps of the above algorithms with respect to the optimal selection have not been verified. For the design of applicable and efficient selection schemes, one important thing is to evaluate the potential performance loss compared to the optimal selection. In order to address the above concerns, the authors of [6] developed a branch-and-bound (BAB) algorithm in order to reduce the complexity of the optimal selection in massive MIMO. It has been shown that BAB achieves significant complexity reduction by comparing to the exhaustive search. However, with the problem scale growing larger, especially when the number of antennas is greater than 128, the BAB algorithm still involves a prohibitive computational complexity, which is also the motivation of this work.

In this paper, we are interested in developing a low-complexity optimal selection algorithm for the very large-scale systems. In contrast to the BAB algorithm [6], we propose to reformulate the optimal antenna selection problem as the shortest path finding on the combinatorial decision tree. In particular, we propose a best-first search antenna selection (BFS-AS) method to find the shortest path efficiently even with a limited hardware memory. We show that the proposed BFS-

AS achieves the exact optimal performance, and meanwhile incurs considerable complexity reduction with comparison to the existing optimality achieving approaches. In particular, when the number of antennas greater than 128, we achieve a four-fold complexity reduction compared with the exhaustive search.

II. Receive Antenna Selection in Massive MU-MIMO System

A. System Model

We consider the uplink of a single-cell massive MU-MIMO system in flat fading environments, where a BS receives data streams from N_u single-antenna users. There are N_r receiving antennas and L ($N_u < L \ll N_r$) dedicated RF chains equipped at the BS. We denote $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_j, \cdots, \boldsymbol{h}_{N_r}]^T$ as the $N_r \times N_u$ full CSI matrix, where $\boldsymbol{h}_j$ represents the channel vector for the j-th receive antenna at the BS. Let $\boldsymbol{x} \in \mathbb{C}^{N_u \times 1}$ be the transmit symbol vector of all users with unit power. The received signal $\boldsymbol{y}$ at the BS can be expressed as

$$\boldsymbol{y} = \sqrt{\rho}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}, \tag{1}$$

where $\boldsymbol{w}$ is the additive white Gaussian noise (AWGN) vector whose elements are random variable with zero-mean and unit variance, and ρ denotes the average signal-to-noise ratio (SNR).

The considered system operates in time division duplex (TDD) mode, meaning that we have identical channels for both uplink and downlink transmission due to channel reciprocity. We assume perfect channel acquisition at the BS [8], and the uplink system spectral efficiency is given by

$$C(\boldsymbol{H}) = \log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{H}^H \boldsymbol{H}\right), \tag{2}$$

where $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, $\det(\cdot)$ gives the matrix determinant, and $(\cdot)^H$ represents the Hermitian transpose. For the uplink transmission, the selection of L out of N_r receiving antennas is needed due to the lack of dedicated RF chains, i.e., $L \ll N_r$. In other words, L out of N_r rows should be chosen from the complete channel matrix $\boldsymbol{H}$ to form a $L \times N_u$ channel sub-matrix. We denote $\boldsymbol{a} = \{a_1, a_2, \cdots, a_j, \cdots, a_L\}$ as the set of selected antenna indices, and let $\mathcal{A}$ denote the set of all possible receive antenna combination for the considered system, and then we have $|\mathcal{A}| = \binom{N_r}{L}$. For a specific receive antenna combination $\boldsymbol{a} \in \mathcal{A}$, we denote $\boldsymbol{H}(\boldsymbol{a})$ as the corresponding sub-matrix. The optimal selection aims to find out the best subset of antennas such that the system spectral efficiency is maximized, which is given by

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a} \in \mathcal{A}} \log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{H}(\boldsymbol{a})^H \boldsymbol{H}(\boldsymbol{a})\right). \tag{3}$$

Obviously, the above problem is NP-hard, and finding the optimal antenna combination requires exhaustive search over the candidate space growing exponentially with the problem scale, which is henceforth infeasible for large-scale MIMO systems.

B. Antenna Selection with Greedy Search

In order to tackle the antenna selection problem in massive MIMO, greedy search based algorithms [2], [3], [6] were proposed with low computational complexity. The greedy search based algorithm works by either greedily removing or selecting one antenna at each step, resulting in that the selection is sub-optimal. However, we will hereby review the later one since it has lower computational complexity [8].

For the n-th step, we denote $\boldsymbol{a}_n = \{a_1, a_2, \cdots, a_n\}$ as the set of currently selected antenna indices, and let $\boldsymbol{H}(\boldsymbol{a}_n)$ be the corresponding channel matrix with n rows selected from the full channel, and the corresponding channel capacity is $C(\boldsymbol{a}_n) = \log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{H}(\boldsymbol{a}_n)^H \boldsymbol{H}(\boldsymbol{a}_n)\right)$. For the next $(n+1)$-th step, consider that the j-th row $\boldsymbol{h}_j \subseteq \widetilde{\boldsymbol{H}}(\boldsymbol{a}_n)$ will be selected from the remaining rows $\widetilde{\boldsymbol{H}}(\boldsymbol{a}_n) = \boldsymbol{H} \backslash \boldsymbol{H}(\boldsymbol{a}_n)$. Then, the currently selected antenna indices becomes $\boldsymbol{a}_{n+1} = \{a_1, a_2, \cdots, a_n, a_j\}$, and the corresponding channel matrix becomes $\boldsymbol{H}(\boldsymbol{a}_{n+1}) = \begin{bmatrix} \boldsymbol{H}(\boldsymbol{a}_n) \\ \boldsymbol{h}_j \end{bmatrix}$, and thereby the channel capacity for the $(n+1)$-th step is given by

$$\begin{aligned} C(\boldsymbol{a}_{n+1}) = &\log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{H}(\boldsymbol{a}_n)^H \boldsymbol{H}(\boldsymbol{a}_n) + \rho \boldsymbol{h}_j^H \boldsymbol{h}_j\right) \\ &= C(\boldsymbol{a}_n) \\ + \log_2 \det &\left(\boldsymbol{I}_{N_u} + \rho \underbrace{\left(\boldsymbol{I}_{N_u} + \boldsymbol{H}(\boldsymbol{a}_n)^H \boldsymbol{H}(\boldsymbol{a}_n)\right)^{-1}}_{\boldsymbol{A}_n} \boldsymbol{h}_j^H \boldsymbol{h}_j\right) \\ &\overset{(a)}{=} C(\boldsymbol{a}_n) + \underbrace{\log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{h}_j^H \boldsymbol{A}(\boldsymbol{a}_n) \boldsymbol{h}_j\right)}_{\Delta_{j,n}}, \end{aligned} \tag{4}$$

where $\boldsymbol{A}(\boldsymbol{a}_n) \triangleq \left(\boldsymbol{I}_{N_u} + \boldsymbol{H}(\boldsymbol{a}_n)^H \boldsymbol{H}(\boldsymbol{a}_n)\right)^{-1}$, and step (a) holds according to the Sylvester's determinant identity. It should be noted that the increment $\Delta_{j,n} \triangleq \log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{h}_j^H \boldsymbol{A}(\boldsymbol{a}_n) \boldsymbol{h}_j\right) \geq 0$ is a non-negative scalar due to the fact that $\boldsymbol{A}(\boldsymbol{a}_n)$ is positive definite. This implies that (4) monotonically increases along with more antennas to be selected. Therefore, a greedy search algorithm will typically aim to maximize the increment $\Delta_{j,n}$ at each step to find the solution of problem (3). Since the increment $\Delta_{j,n}$ is dominated by the quadratic forms $\boldsymbol{h}_j^H \boldsymbol{A}(\boldsymbol{a}_n) \boldsymbol{h}_j$, the greedy search based algorithm is also well-known as norm-based selection. As such greedy search algorithms only carry the local optimal at each step, it is clear that the resulting solution is sub-optimal.

As has been pointed out in [3], [6], the greedy search based algorithm is guaranteed to be optimal under certain conditions. However, those conditions often do not hold for practical systems, especially when the system scale is very large. In this case, the greedy search algorithm is not optimal and the performance may degrade severely [6]. In further, for the developing of low-complexity sub-optimal antenna selection algorithms, a feasible optimal antenna selection is also necessary for the evaluation of the potential performance loss in very large-scale systems. Hence, researchers in [6] proposed to employ BAB method to reduce the complexity of finding the optimal solution. However, the simulation results show that the complexity of the proposed BAB method is still very high in very large-scale systems, which motivates us to develop a low-complexity optimal antenna selection algorithm for very large-scale systems.

III. Proposed Fast Optimal Antenna Selection Algorithm

A. Tree Construction

In order to find the optimal solution efficiently, we first need to construct a search tree for problem (3). To this end, we first notice the following result.

Lemma 1. *[6] Given* $\Delta_{j,n} \triangleq \log_2 \det\left(\boldsymbol{I}_{N_u} + \rho \boldsymbol{h}_j^H \boldsymbol{A}(\boldsymbol{a}_n)\boldsymbol{h}_j\right)$ *as described in (4), the upper bound of* $\Delta_{j,n}$ *is given by*

$$\Delta_{j,n} \leq Z_n = \max_{j\in\mathcal{I}_n} \log_2(1+\rho\|\boldsymbol{h}_j\|_F^2), \quad n = 0,1,\cdots,L-1 \tag{5}$$

where $\|\boldsymbol{h}\|_F$ *denotes the Frobenius norm, and* $\mathcal{I}_n$ *denotes the set of candidate antenna indices at the* n*-th selection step.*

According to the above lemma, it is clear that for a complete selection at the L-th step, we have the upper bound on the accumulative increments as

$$C(\boldsymbol{a}_L) = \sum_{n=0}^{L-1} \Delta_{j,n} \leq C_U, \tag{6}$$

where $C_U = \sum_{n=0}^{L-1} Z_n$. Clearly, C_U is fixed for every $\boldsymbol{H}$, which implies that it is irrelevant to the selected antenna combination and only relies on the channel realization. Based on this fact, we propose an alternative objective function as

$$G(\boldsymbol{a}_L) = C_U - C(\boldsymbol{a}_L) \tag{7}$$

By using (4), we have the recursive form of the new objective function as

$$\begin{aligned} G(\boldsymbol{a}_{n+1}) &= C_U - C(\boldsymbol{a}_{n+1}) \\ &= \sum_{k=0}^{n-1} Z_k - C(\boldsymbol{a}_n) + \underbrace{Z_n - \Delta_{j,n}}_{B_{j,n}} \\ &= G(\boldsymbol{a}_n) + B_{j,n}, \end{aligned} \tag{8}$$

where the least Z_n that satisfies $Z_n \geq \Delta_{j,n}$ will be assigned to the n-th step from the remaining Z_n's, and eventually all L terms will be assigned to L steps.

In this case, we have $B_{j,n} \triangleq Z_n - \Delta_{j,n} \geq 0$, and $B_{j,n}$ can be interpreted as the potential capacity loss of selecting antenna index j at step n. Meanwhile, the new objective function (8) remains a monotonically increasing function, which indicates that problem (3) can be reformulated as

$$\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}\in\mathcal{A}} \left\{C_U - C(\boldsymbol{a})\right\}. \tag{9}$$

In this manner, the optimal subset of antennas leading to the minimal potential capacity loss actually also leads to the exact optimal solution of problem (3). As contrast to the objective function introduced in [6], this objective function

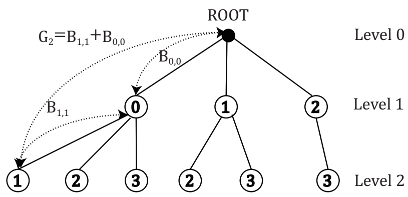

Fig. 1. Example for the combinatorial tree with $N_r = 4$ and $L = 2$.

allows us to convert problem (3) as the problem of finding the shortest path on the tree of (8). As show in Fig. 1, in the resulting combinatorial search tree, each branch represents a possible selection of antenna indices, and the branch cost for selecting antenna j at step n is thereby defined by $B_{j,n}$. Hence, the accumulative cost of a complete path gives the total potential capacity loss, and finding the shortest path on the tree will eventually obtain the optimal subset of antennas which maximizes the channel capacity.

B. Optimal Selection with Efficient Best-First Tree search

In order to find the shortest path efficiently, we introduce an efficient best-first antenna selection (BFS-AS) algorithm algorithm, which can use the best-first search strategy [11] to achieve the exact optimal performance and meanwhile only requires a fixed space complexity. Before presenting the details of the proposed algorithm, it is necessary to introduce the following important concepts:

- `MEMORY` is a self-balancing binary search tree (a.k.a. AVLTree) with a fixed size, in which the elements are organized according to their costs to support high frequency queries.
- If a node has been visited during the expansion of its parent, then this node is *generated*.
- If a node occupies space in `MEMORY`, then this node is *in memory*.
- If all the children of a node are in memory, then this node is *completely expanded*.

With the above concepts, we are now ready to introduced the proposed algorithm. In particular, the pseudo code of the proposed BFS-AS algorithm is presented in Algorithm 1, and we separate the internal functions, and merge them into Algorithm 2 for the ease of presentation.

As shown in Algorithm 1, BFS-AS works by exploring the currently best combination of each loop. If there is no enough space for further expansion, the algorithm will forget the most unpromising leaf combination. In this manner, BFS-AS can limit the usage of memory space, and still has the capability to find out the shortest path. This procedure always starts from the dummy root, and it actually maintains and updates a partially expanded combinatorial tree for each iteration. Since only the currently best combination will be selected for expansion, all the other combinations with greater G-cost than the last generated complete path will be pruned. In particular, as shown in Line 5 of Algorithm 1, BFS-AS will expand only

Algorithm 1 BFS-AS Algorithm

Input: Full channel matrix $\boldsymbol{H}$
Output: Selected antenna combination $\boldsymbol{a}$

Push the dummy root $\boldsymbol{a}_0 = \varnothing$ into MEMORY;
loop
  $\boldsymbol{a}_k \leftarrow$ *deepest least-G-cost combination* in MEMORY;
  if $\boldsymbol{a}_k$ is a complete selection ($k = L$) **then**
    return $\boldsymbol{a}_k$;
  end if
  $\boldsymbol{a}_{k+1} \leftarrow$ next (not generated or best forgotten) child combination of $\boldsymbol{a}_k$
  Mark $\boldsymbol{a}_{k+1}$ as a generated child of $\boldsymbol{a}_k$;
  BACKPROPAGATION($\boldsymbol{a}_k$);
  MANIPULATEMEMORY();
  Push $\boldsymbol{a}_{k+1}$ into MEMORY;
  if $\boldsymbol{a}_k$ is completely expanded **then**
    Pop $\boldsymbol{a}_k$ from MEMORY;
  end if
end loop

Algorithm 2 Internal Functions for the BFS-AS Algorithm

function BACKPROPAGATION($\boldsymbol{a}_k$)
  if all the children of $\boldsymbol{a}_k$ have been generated **then**
    $\boldsymbol{a}_{k+1} \leftarrow$ least-G-cost child from the generated and forgotten children of $\boldsymbol{a}_{k+1}$;
    if $G(\boldsymbol{a}_{k+1}) \neq G(\boldsymbol{a}_k)$ **then**
      $G(\boldsymbol{a}_k) = G(\boldsymbol{a}_{k+1})$;
      BACKPROPAGATION($\boldsymbol{a}_k$'s parent);
    end if
  end if
end function

function MANIPULATEMEMORY()
  if MEMORY is full **then**
    Pop *shallowest highest-cost leaf combination* $\boldsymbol{a}_j$ from MEMORY;
    Mark $\boldsymbol{a}_j$ as a forgotten combination of its parent;
    if $\boldsymbol{a}_j$'s parent combination is not in MEMORY **then**
      Push $\boldsymbol{a}_j$'s parent combination into MEMORY;
      MANIPULATEMEMORY();
    end if
  end if
end function

once completed path during the search, which is different from the BAB algorithm introduced in [6]. This difference ensures BFS-AS can visit much less combinations than the BAB method, and thus it can run faster in large-scale systems.

It should be emphasized that the optimality does not rely on the size of MEMORY, as long as the least required space for expanding a complete path is provided. For different sizes of MEMORY, BFS-AS still explores the same combination set. However, depending on the size of MEMORY [11], some combinations may be visited more than once. This implies that it is always better to have a larger MEMORY, as frequent MEMORY manipulation should be avoided. In particular, BFS-AS will work exactly as the same as the best-first style search with infinite MEMORY.

C. Complexity

Basically, the time complexity of the proposed BFS-AS algorithm for exploring one combination on the tree is the same as the greedy search [3] and BAB [6], i.e. $\mathcal{O}(N_r N_u)$. The total complexity of BFS-AS is determined by the $\mathcal{O}(N_r N_u N_v)$, where N_v is number of visited nodes on the tree. In particular, N_v varies for different channel realizations, and henceforth we will measure the expected complexity with Monte-Carlo simulations in the next section. As a contrast, the greedy search and exhaustive search will explore a fixed number of nodes, which are $N_v = \frac{N_r!}{(N_r-L)!}$ and $N_v = L\binom{N_r}{L}$ respectively.

IV. SIMULATION

In this section, we present simulation results to verify the effectiveness of the proposed BFS-AS algorithm. The competing algorithms include the low-complexity greedy search algorithm [3], the BAB algorithm [6], and the proposed BFS-AS algorithm. It should be noted that both BAB and BFS-AS are optimal, while the greedy search is sub-optimal. For the complexity comparison, we will measure the time complexity according to the average number of visited nodes, since all the aforementioned algorithms are search based algorithms. In the simulations, we assume that perfect channel acquisition at the BS. In addition, $N_u = L = 4$, N_r varies from 16 to 128 with a step size of 16, and the SNR is set to 20 dB. Moreover, the simulation results are taken from the average of 10^4 trials, and we limit the maximum memory usage of BFS-AS at the level of 100 MB.

Fig. 2 demonstrates the capacity comparison versus the number of receive antennas N_r for the aforementioned algorithms, where $N_u = L = 4$ and SNR is set to 20 dB. We can find from this figure that the proposed BFS-AS reaches the exact optimal performance, as it has the same capacity as the optimal BAB algorithm. In addition, we can conclude from Fig. 2 that the performance gap between the sub-optimal greedy search and BFS-AS enlarges with the increasing number of receive antennas. Specifically, when $N_r = 128$, greedy search has a noticeable spectral efficiency loss of around 0.28 bps/Hz with comparison to BFS-AS. This is because with the increasing N_r, the greedy search algorithm still explores a fixed number of antenna combinations. As the explored portion decreases significantly with the increasing N_r, the performance gap between greedy search and BFS-AS also enlarges, especially for the very large-scale systems with $N_r \geq 128$.

Fig. 3 dipicts the complexity comparison versus the number of receive antennas N_r for the aforementioned algorithms, where $N_u = L = 4$ and SNR is set to 20 dB. Since the competing algorithms have exactly the same complexity for visiting a single node, it is accurate to evaluate their complexities according to the number of visited nodes. It

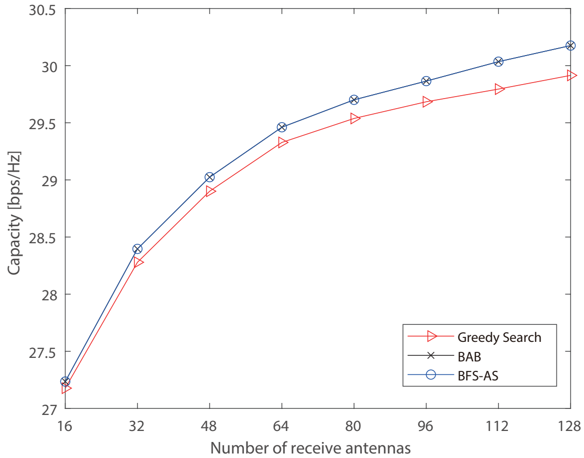

Fig. 2. Capacity comparison versus N_r with $N_u = L = 4$ and SNR $= 20$ dB.

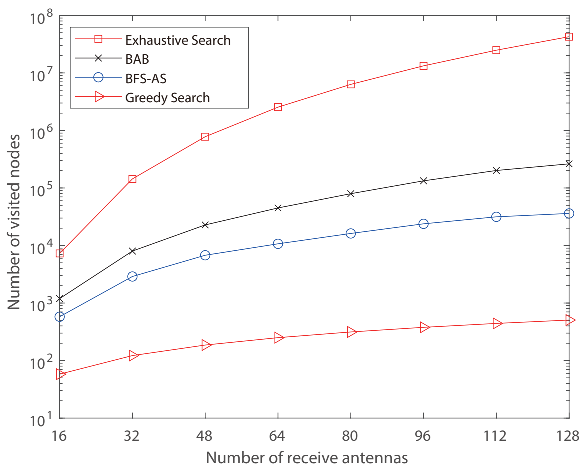

Fig. 3. Complexity comparison versus N_r with $N_u = L = 4$ and SNR $= 20$ dB.

can be concluded from Fig. 3 that considerable complexity reduction is achieved for both BAB and the proposed BFS-AS with respect to the exhaustive search. This is because both BAB and BFS-AS has the capabity to avoid visiting many unpromising nodes. In principle, BAB and BFS-AS actually achieve an intermediate complexity between the low-complexity greedy search and the prohibitive exhaustive search, while still maintaining the optimal performance. In particular, we can find from Fig. 3 that the proposed BFS-AS visits significant lower nodes with comparison to BAB, especially when N_r enlarges, and the complexity gap between BFS-AS and BAB enlarges with larger N_r. Specifically, when $N_r = 128$, BFS-AS visits 87% less nodes by comparing to BAB, and BFS-AS reduces the number of visited nodes by 4 orders of magnitude by comparing to exhaustive search. This is because BFS-AS imposes a best-first style search such that it can reduce more unpromising nodes than BAB. In conclusion, the proposed BFS-AS not only reaches the exact optimal performance, but also has lower average complexity.

V. Conclusion

In this paper, we have investigated the optimal antenna selection problem for massive MU-MIMO. In order to find a feasible optimal antenna selection approach for very large-scale systems, we have proposed an alternative objective function based on the upper bound of channel capacity. By utilizing the proposed objective function, the optimal antenna selection problem can be converted as a shortest path finding on the resulting combinatorial tree. In addition, we have further proposed an efficient best-first search based algorithm, namely BFS-AS, for the fast shortest path finding. Simulation results showed that the proposed BFS-AS not only reaches the exact optimal performance, but also achieves significantly lower complexity with comparison to the existing optimal algorithms including BAB and exhaustive search.

As the existing optimal methods are infeasible for very large-scale systems, it is still very difficult to evaluate the performance loss for the algorithm design in such scenarios. Therefore, we believe the proposed BFS-AS is of importance for such evaluations, and it can provide good insights for the future low-complexity high-performance antenna selection algorithm design.

Acknowledgement

This work is supported by the Luxembourg National Funds, project RUTINE (ref. FNR/C22/IS/17220888/RUTINE) and project ASWELL (ref. FNR/C19/IS/13718904/ASWELL

References

[1] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive mimo: Benefits and challenges," *IEEE journal of selected topics in signal processing*, vol. 8, no. 5, pp. 742–758, 2014.

[2] A. Gorokhov, "Antenna selection algorithms for mea transmission systems," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 2002, pp. III–2857.

[3] M. Gharavi-Alkhansari and A. B. Gershman, "Fast antenna subset selection in mimo systems," *IEEE transactions on signal processing*, vol. 52, no. 2, pp. 339–347, 2004.

[4] A. F. Molisch, M. Z. Win, Y.-S. Choi, and J. H. Winters, "Capacity of mimo systems with antenna selection," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1759–1772, 2005.

[5] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[6] Y. Gao, H. Vinck, and T. Kaiser, "Massive mimo antenna selection: Switching architectures, capacity bounds, and optimal antenna selection algorithms," *IEEE Transactions on signal processing*, vol. 66, no. 5, pp. 1346–1360, 2017.

[7] Z. Liu, Y. Yang, F. Gao, T. Zhou, and H. Ma, "Deep unsupervised learning for joint antenna selection and hybrid beamforming," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1697–1710, 2022.

[8] J. Chen, S. Chen, Y. Qi, and S. Fu, "Intelligent massive mimo antenna selection using monte carlo tree search," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5380–5390, 2019.

[9] K. He, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Learning-based joint channel prediction and antenna selection for massive mimo with partial csi," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 178–183.

[10] T. X. Vu, S. Chatzinotas, V.-D. Nguyen, D. T. Hoang, D. N. Nguyen, M. Di Renzo, and B. Ottersten, "Machine learning-enabled joint antenna selection and precoding design: From offline complexity to online performance," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3710–3722, 2021.

[11] S. J. Russell, "Efficient memory-bounded search methods," in *Proc. European Conference on Artificial Intelligence (ECAI)*, 1992, pp. 1–5.