Smart Sorting in Massive MIMO Detection

Andrey Ivanov, Dmitry Yarotsky, Maria Stoliarenko, Alexey Frolov
Skolkovo Institute of Science and Technology
Moscow, Russia
AN.Ivanov@skoltech.ru, D.Yarotsky@skoltech.ru, M.Stoliarenko@skoltech.ru, Al.Frolov@skoltech.ru

***Abstract*—In this paper, we proposed a discrete sorting optimization approach for uplink channel of Massive Multiple Input, Multiple Output (MIMO) system. The algorithm includes users (UEs) sorting before QR decomposition (QRD) and sorting-reduced (SR) K-best detector for 48x64 MIMO uncoded systems. Simulation results show that detector losses is about 1dB to a Maximum Likelihood (ML) detection in low detector complexity.**

The UEs sorting is required to sort diagonal elements of the R matrix in ascending order to avoid error propagation in multi-user (MU) scenario. Fast and low complexity method of online discrete optimization is used to find the loss function minimum. Sorting tracking is proposed, so that the pre-sorted interpolated R matrix is used for further sorting optimization, resulting in low sorting complexity. The proposed sorting demonstrates huge performance gain compare to a common power-based one. Simulation results in 5G QuaDRiGa channel are presented.

SR-K-best detector is a variant of K-best detector. The SR-K-best with (K,S,p) parameters results in significant losses in scenarios with high correlated users, therefore we proposed a new structure (K,S,p,v,q) of the SR-K-best algorithm and a new discrete optimization method to increase performance. Discrete stochastic optimization was done offline in QuaDRiGa channel to find optimal (K,S,p,v,q) parameters for fixed detector structure.

I. Introduction

MU-MIMO detection, well described in [1] and [2], is a method of combining antennas digital signal to increase user signal power in uplink channel of base station. It is a key technology for future 5G systems [3], which provides extremely high system capacity. Switching from a common MIMO to Massive MIMO requires efficient and fast signal processing algorithms since implementation complexity is the main concern with a very large number of antennas (e.g., tens or hundreds). For MIMO detection there is always a trade-off between performance and complexity. Intuitively, computational resources should be utilized more efficiently by performing optimal detection. A typical MIMO detection scenario is shown in Fig. 1.

Maximum Likelihood (ML) detector is the optimal detector because it checks all possible combinations of TX symbols and chooses one with the minimum Euclidean distance. However, the complexity grows exponentially with high modulation order (QAM16 or higher) or large number of TX antennas (say > 16). The ML estimation is given by [2]:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

$$\boldsymbol{x}_{ML} = \arg\min \left\| \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x} \right\|,$$

The research was carried out at Skoltech and supported by the Russian Science Foundation (project no. 18-19-00673).

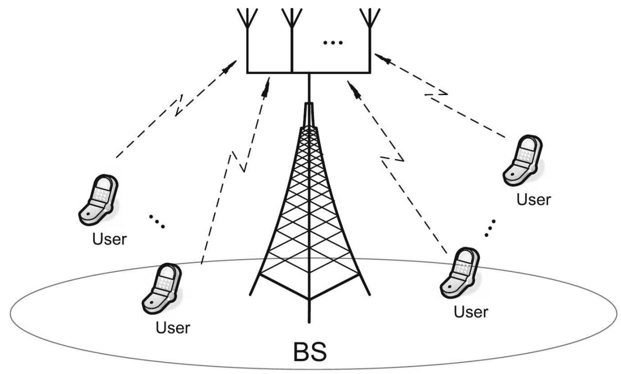

Fig. 1: MIMO detection scenario

where $\boldsymbol{y}$ is the frequency domain received signal vector of size N for each subcarrier; N is the number of receive antennas; $\boldsymbol{H}$ is the propagation channel matrix of $[N \times M]$ size for each subcarrier, $\boldsymbol{x}$ is the transmitted signal vector of M size; M is the number of single antenna UEs; $\boldsymbol{n}$ is an additive noise vector; $\boldsymbol{x}_{ML}$ is the ML estimation of x. Complexity of ML detection exponentially increases with the number of transmit antennas.

Minimum mean square error (MMSE) estimator is a baseline linear solution of the detection problem, described in [2] and [4]. However, huge performance losses in MU-MIMO scenario limit the MMSE application, and a new algorithm is required. The MMSE estimation is calculated as:

$$\boldsymbol{x}_{MMSE} = \boldsymbol{W}\boldsymbol{y},$$
$$\boldsymbol{W} = \frac{\boldsymbol{H}^H}{\boldsymbol{H}^H\boldsymbol{H} + \frac{\sigma_n^2}{\sigma_x^2}\boldsymbol{I}},$$

where $\boldsymbol{x}_{MMSE}$ is the MMSE estimation of $\boldsymbol{x}$; $\boldsymbol{W}$ is the weight matrix; σ_n^2 is the receiving antenna noise power; σ_x^2 is a transmitted UE power; $\boldsymbol{I}$ is $[M \times M]$ eye matrix. Equation can be calculated in simplified form by using an extended channel matrix:

$$\boldsymbol{W} = \frac{\boldsymbol{H}_{ext}^H}{\boldsymbol{H}_{ext}^H\boldsymbol{H}_{ext}},$$
$$\boldsymbol{H}_{ext} = \begin{bmatrix} \boldsymbol{H}^T & \frac{\sigma_n}{\sigma_x}\boldsymbol{I} \end{bmatrix}^T, \tag{2}$$
$$\boldsymbol{y}_{ext} = \begin{bmatrix} \boldsymbol{y}^T & \boldsymbol{0} \end{bmatrix}^T,$$

MMSE with ordered successive interference cancellation (OSIC) MIMO detector is performed from the QRD of the permuted channel matrix $\boldsymbol{H}_{perm}$, which is defined as in [5], [6] and [7]:

$$\boldsymbol{y}_{ext} = \boldsymbol{H}_{perm}\boldsymbol{x}_{perm} + \boldsymbol{n},$$
$$\boldsymbol{x}_{perm} = \boldsymbol{P}\boldsymbol{x},$$
$$\boldsymbol{H}_{perm} = \boldsymbol{P}\boldsymbol{H}_{ext}, \tag{3}$$

where matrix $\boldsymbol{H}_{ext}$ is calculated by equation (2) and used instead of $\boldsymbol{H}$ for regularization reasons. QR factorization plays a key role in the OSIC detection [6]:

$$\boldsymbol{H}_{perm} = \boldsymbol{Q}\boldsymbol{R},$$
$$\boldsymbol{Q}^H\boldsymbol{y}_{ext} = \left(\boldsymbol{Q}^H\boldsymbol{Q}\right)\boldsymbol{R}\boldsymbol{x}_{perm} + \boldsymbol{Q}^H\boldsymbol{n},$$
$$\boldsymbol{Q}^H\boldsymbol{y}_{ext} = \boldsymbol{R}\boldsymbol{x}_{perm} + \boldsymbol{Q}^H\boldsymbol{n},$$

where $\boldsymbol{R}$ is a $[(N+M)\times M]$ upper triangular matrix; $\boldsymbol{P}$ is the permutation matrix, required to reorder UEs before the SIC detection. Iterative detection starts from $\boldsymbol{x}_{perm}(M)$ amplitude detection and stops with $\boldsymbol{x}_{perm}(1)$ calculation due to upper triangle matrix $\boldsymbol{R}$ property. After that initial vector $\boldsymbol{x}$ can be calculated as $\boldsymbol{x} = \boldsymbol{P}^T\boldsymbol{x}_{perm}$. MMSE-OSIC detection demonstrates improved performance compared to the MMSE detection, but the improvement is limited due to error propagation issues, caused by non-ideal permutation (or sorting). Most known sorting methods are too complex or based on power reordering, resulting in performance losses in correlated channel.

Lattice reduction aided (LRA) type of methods used in [1] and [8] are pre-processing procedures that operate on the channel matrix $\boldsymbol{H}$ only rather than pure detection methods that operate on the received data vector $\boldsymbol{y}$ as well. The LRA MIMO detection algorithms build upon the idea of converting an ill-conditioned problem into an equivalent well-conditioned problem via a linear transform $\boldsymbol{T}$ that fulfills certain conditions. Lattice reduction reorganizes the extended channel matrix to more orthogonal one, lowering the likelihood of noise propagation:

$$\boldsymbol{y} = \boldsymbol{H}_{ext}\boldsymbol{T}\boldsymbol{T}^{-1}\boldsymbol{x} + \boldsymbol{n} = \boldsymbol{H}_{LR}\boldsymbol{z} + \boldsymbol{n},$$
$$\boldsymbol{z} = \boldsymbol{T}^{-1}\boldsymbol{x},$$
$$\boldsymbol{H}_{LR} = \boldsymbol{H}_{ext}\boldsymbol{T},$$
$$\boldsymbol{x} = a\boldsymbol{P}^T\left[round\left(\boldsymbol{T}\boldsymbol{z}\right) + \boldsymbol{1}\frac{1}{2}\left(1+i\right)\right],$$

where $\boldsymbol{H}_{LR}$ is the lattice-reduced channel matrix; $\boldsymbol{T}$ is the uni-modular matrix; a is the scaling coefficient ($a = \frac{2}{\sqrt{10}}$ for QAM16 and $a = \frac{2}{\sqrt{42}}$ for QAM64) . Thus, many LRA detectors use function round() to slice the invalid vector $\boldsymbol{T}\boldsymbol{z}$ to the nearest point of $\boldsymbol{x}$. Once the problem is well-conditioned, common MMSE or MMSE-OSIC detectors can be used to achieve near-optimal performance. The complex Lenstra–Lenstra–Lovász (cLLL) algorithm, described in [1], [6] and [8], is a well-known polynomial time LRA algorithm. However, it is inefficient for direct implementation because of several problems:

1) variable extra complexity of cLLL implementation;
2) slicing (rounding to the nearest point as shown in [1] and [5]) and candidates list generation in LR domain since $\boldsymbol{T}$ transformation results in non-rectangle constellation and fixed candidates numeration is less efficient;
3) log-likelihood ratio (LLR) calculation in LR domain because most modern decoders require soft decision;
4) interpolation-based cLLL (i.e. how to interpolate $\boldsymbol{T}$ matrix from pilot symbol to the current one, such approach is described in [9] for QRD without LRA).

Many researchers have proposed solutions to decrease LRA complexity or fix the processing iteration of the LR algorithm in their hardware implementation, for example [1] and [6], but LRA still requires a lot of computations and, moreover, performance gap between ML and LRA solution is still big in highly correlated UEs scenarios. Nevertheless, most researchers still ignore problems 1-4, only considering performance gain.

Using MMSE or MMSE-OSIC MIMO detector after LR cannot achieve satisfying performance for 48×64 MIMO system. Therefore, the processing K-best detector can be utilized to achieve satisfying performance in adoptable complexity [1]. Although the proposed distributed K-best (D-K-best, [8]) algorithm greatly reduces the sorting complexity compared with a common breadth-first K-best algorithm, the D-K-best also results in significant performance losses in correlated channel. In [1] a variant of D-K-best detector called SR-K-best algorithm was proposed. Sorting the best K survivors from KM ($M = 4$) candidates is reduced to sorting the best S candidates while other $K - S$ survivors are called "most expected" and selected without sorting. Thus, paper [1] proposes an SR-K-best with index $(K, S, \boldsymbol{p})$ to reduce the sorting operation from K to S times. The vector $\boldsymbol{p}$ defines positions of the "most expected" candidates. However, its architecture is not optimal in candidates selection for the next iteration step.

In addition, the MIMO detection problem can be reformulated as how to achieve best performance in the fixed complexity? Alternatively, how to find optimal architecture in fixed complexity? A general approach to this problem is a machine learning application. Machine learning asymptotically converges to ML solution and can increase performance in the receiver. There are many machine learning approaches to MIMO detection problem: support vector regression (SVR) is proposed in [10], convolutional neural networks (CNN) are proposed in [11] and [12]. Authors achieve performance gain in high correlated scenarios and their solution requires less calculations. However, all simulations done with modulation order up to QPSK, since phase modulation fits to neural networks architecture. With QAM16 or higher the machine learning algorithm complexity grows significantly. Another problem appears when in uplink channel there are several users with different modulation scheme, for example, N_1 UEs with QPSK, N_2 UEs with QAM16 and N_3 UEs with QAM64. It is clear that the number of combinations $\left[\begin{array}{ccc} N_1 & N_2 & N_3 \end{array}\right]$ is rather big when $N_i \in [1...64]$ and to learn all such scenarios for each signal-to-noise ratio (SNR) and for each correlation between UEs the CNN should have huge size. Therefore, the advantage of direct machine learning application in MIMO detection is not evident; however, learning ideas could be useful to optimize existing algorithms structures. Our simulations with CNN deep learning in QuaDRiGa channel show that learning error doesn't go down when the number of layers exceeds the number of UEs. Therefore, CNN demonstrates performance gain in MIMO detection due to clever iterative OSIC.

In this paper, we show by simulations in QuaDRiGa channel that MU-MIMO detection is very sensitive to sorting operation and propose smart sorting optimization. In fact, clever UE sorting before QRD and candidates sorting in SR-K-best stage demonstrate final performance similar to LRA application, but in much less complexity, and the proposed solution is more feasible in application.

II. Simulation tool

QuaDRiGa, short for QUAsi Deterministic Radio channel GenerAtor [13], is a Matlab software used for generating realistic radio channel impulse responses for system-level simulations of mobile radio networks. These simulations are used to determine the performance of new technologies in order to provide an objective indicator for the standardization process in bodies like the third generation partnership program 3GPP. Besides being a fully-fledged three dimensional geometry-based stochastic channel model, QuaDRiGa contains a collection of channel models along with novel modelling approaches, which provide features to enable quasi-deterministic multi-link tracking of users movements in changing environments. QuaDRiGa contains a couple of new features and is furthermore calibrated against 3GPP channel models like 3GPP-3D and the latest New Radio channel model. The QuaDRiGa approach can be understood as a statistical raytracing model. In our simulation we used non-line of sight (NLOS) scenario with 48 single antenna UEs moving with averaged speed of 3km per hour and 64 receiving antennas of BS (MIMO 48×64). A short fragment (magnitude spectrum) of generated channel model is shown in Fig. 2 for 600 subcarriers in 90 symbols.

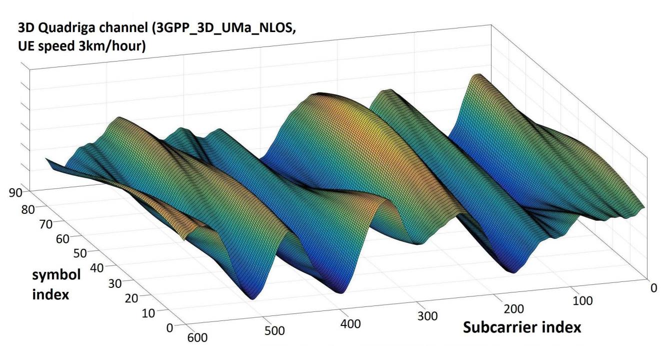

Fig. 2: Magnitude spectrum of QuaDRiGa channel

III. Detection approach

A functional scheme of the proposed approach is shown in Fig. 3. It consists of pre-processing and processing parts. Preprocessing is required to calculate sorted QRD in two steps.

***Step* 1**: QRD interpolation as described in [14].

For QRD calculation in the MIMO system, we have to estimate the channel response of each subcarrier and then perform the QRD for each subcarrier. In practice, the interpolation-based QRD only computes the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrixes for the pilot subcarriers to reduce a computational complexity. Then, the $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the data subcarriers are interpolated from those of the pilot subcarriers.

***Step* 2**: UE sorting (strings permutation in matrix $\boldsymbol{H}_{ext}$) to find the $\boldsymbol{P}$ matrix for equation (3).

UE sorting problem is well-known and there are many approaches to overcome it, for example a post-sorting algorithm and pre-sorting solution are analyzed in [6]. However, most methods require many computation resources or result in poor performance. We propose to use $\boldsymbol{P}$ matrix from QRD of the pilot symbols as the first step of $\boldsymbol{P}$ matrix calculation for the data symbol. Therefore, sorting tracking is considered. The $\boldsymbol{P}$ matrix slightly changes from one symbol to another and a low sorting complexity is required to update it. Loss function $L = L\{\mathrm{diag}\,(\boldsymbol{R})\}$ of diagonal entries of interpolated $\boldsymbol{R}$ matrix is optimized to guarantee the least number of matrix $\boldsymbol{P}$ updates (low complexity updates) from one symbol to another without performance losses. Processing part is represented by

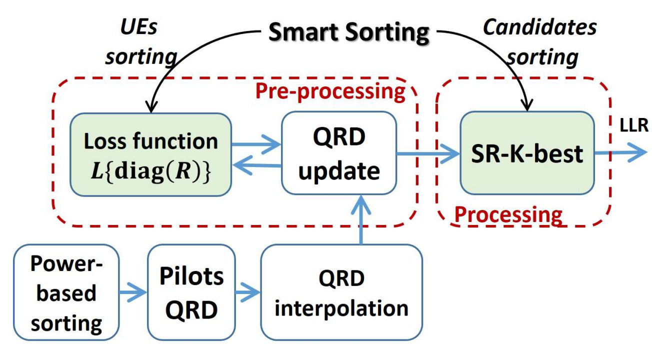

Fig. 3: Functional MIMO detection scheme

optimized SR-K-best iterative detection. First of all, $(K, S, \boldsymbol{p})$ parameters were optimized in QuaDRiGa channel of 48 × 64 MIMO system. Number of users = 48 is chosen as the max realistic number of active UEs per subcarrier in 64 RX antennas system. Besides $(K, S, \boldsymbol{p})$ optimization, a new vectors $(\boldsymbol{v}, \boldsymbol{q})$ are proposed and optimized. The vector $\boldsymbol{v}$ defines a set of sorting child nodes (i.e. a set for S sorted candidates search); while the vector $\boldsymbol{q}$ defines location of S sorted candidates in the final composed list of K candidates for the next detection iteration. Simulation demonstrates extra gain without any additional computations.

IV. SMART UE SORTING BEFORE QRD

Let $\boldsymbol{H}$ be a $N = 64$ by $M = 48$ matrix of full rank M. We consider the QRD of matrix $\boldsymbol{H}_{perm} = \boldsymbol{P}\boldsymbol{H}_{ext}$ according to equations (2) and (3) with permuted strings, i.e. equation $\boldsymbol{H}_{perm} = \boldsymbol{QR}$. We assume without loss of generality that the matrix $\boldsymbol{R}$ in this decomposition is square. We are interested in finding permutations such that an absolute value of the diagonal entries $|R_{nn}|$ tend to increase as n grows. In general, it is not possible to find a permutation achieving a complete ordering, but we can fix a specific loss function

$$L = L\{\mathrm{diag}\,(\boldsymbol{R})\},$$

penalizing the lack of order, and then consider the problem of optimizing the permutation with respect to this loss function. One obvious choice for the loss function is the number of pairs of diagonal entries with the wrong order:

$$L = \sum_{1\le k<n\le M} (R_{kk} \ge R_{nn}). \tag{4}$$

However, equation (4) is a poor choice for several reasons. First, this loss is excessively (discontinuously) sensitive to the changes in $\boldsymbol{R}$ whenever two diagonal entries become equal, and is entirely insensitive otherwise. Second, even if we perform only some simple permutation on $\boldsymbol{H}_{ext}$ (say exchange a pair of neighboring strings), its effect on the loss is relatively hard to compute: we need to check for each diagonal entry, even if it is unaffected by the permutation, whether any pairs of this entry with the affected diagonal entries change their status (correct/wrong order). A much more sensible choice for the loss function is defined as:

$$L = -\sum_{n=1}^{M} (2n - M - 1)\ln R_{nn} \ . \tag{5}$$

This loss (5) obviously has the *locality* property: if the permutation only involves a range of strings, say $I = [k, k+1, .., m]$, then the change in L after applying the permutation only depends on the old and new diagonal entries of $\boldsymbol{R}$ in this range (the diagonal entries outside of I do not change). On the other hand, this L can be equivalently written as:

$$L = \sum_{1\le k<n\le M} \ln\frac{R_{kk}}{R_{nn}} \ , \tag{6}$$

which is similar to our first considered loss (4), but takes into account the relative magnitude of the diagonal entries.

Consider the permutation of only two neighboring strings of matrix $\boldsymbol{H}_{ext}$, say k and $k+1$. Let R and R' denote the R factors in the QR decompositions of $\boldsymbol{H}_{ext}$ before and after the permutation, respectively. Then $R'_{nn}{=}R_{nn}$ for all $n \neq k, k+1$, while $R'_{k,k} = \sqrt{|R_{k,k+1}|^2 + |R_{k+1,k+1}|^2}$ and $R'_{k+1,k+1} = R_{k,k}R_{k+1,k+1}/\sqrt{|R_{k,k+1}|^2 + |R_{k+1,k+1}|^2}$.

The increase in the loss that this permutation brings then equals $\Delta L = \ln\frac{|R_{k,k+1}|^2 + R_{k+1,k+1}{}^2}{R_{k,k}{}^2}$.

This suggests the following algorithm of finding a locally optimal permutation by multiplying a sequence of above elementary permutations.

1) Given an interpolated pre-sorted QRD for the current symbol (sorting comes from the pilot symbol);
2) We compute the ratios $\frac{|R_{n,n+1}|^2+|R_{n+1,n+1}|^2}{R_{n,n}{}^2}$ for all n and check if there are any less than 1 so that we can decrease the loss;
3) If yes, we perform the respective elementary permutation of neighboring strings, i.e. updating the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ for the current symbol;
4) After that, we update the ratios $\frac{|R_{n,n+1}|^2+|R_{n+1,n+1}|^2}{R_{n,n}{}^2}$ for new matrix $\boldsymbol{R}$;
5) Repeat until there are no ratios less than 1.

Let us estimate the complexity of one step of this algorithm. Updating the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ requires $O(N)$ arithmetic operations. Updating the ratios $\frac{|R_{n,n+1}|^2+|R_{n+1,n+1}|^2}{R_{n,n}{}^2}$ requires $O(2)$ operations, since we only need to recompute it at $n = k-1$ and $n = k+1$, where k corresponds to the elementary permutation of the previous step. Checking if there are ratios less than 1 requires $O(M)$ comparisons. Thus, the total number of one-step comparisons is $O(N+M)$, i.e. $O(N)$ arithmetic operations. At each step of the algorithm, we can ensure without changing the $O(N)$ complexity that we choose n with the lowest currently available ratio, thus obtaining the optimal gain. This can be done simply by minimizing the ratios over all n's at each step or, more efficiently, by maintaining (and keeping updated between steps) the sorted list of ratios smaller than 1.

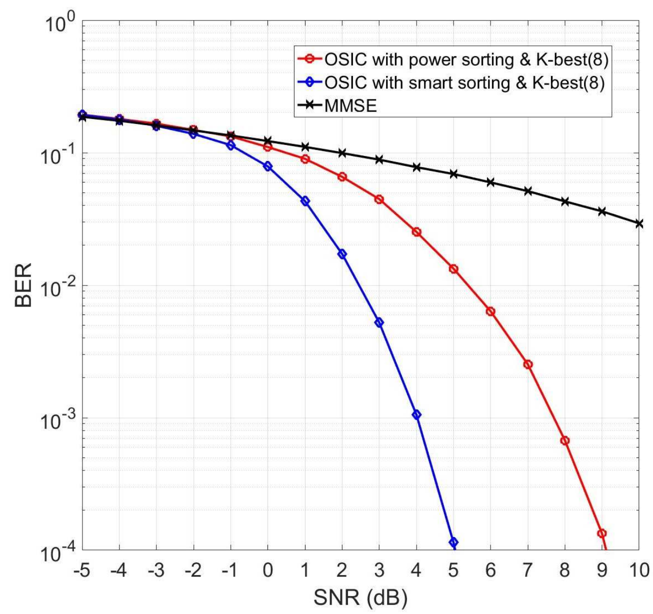

Fig. 4: BER comparison of smart and power-based orderings in 48 × 64 uncoded MIMO system with QAM16

As a crude estimate of the number of steps, note that any permutation of M strings can be written as not more than $\frac{M^2}{2}$ elementary permutations of pairs of neighboring strings. However, in our procedure the optimization of the ordering starts with an already pre-sorted matrix $\boldsymbol{H}_{ext}$. In addition, our optimization (6) only finds a local optimum, which typically does not need a long chain of elementary permutations. As a result, the number of optimization steps is small, so the overall complexity of sorting optimization is low.

Comparison results of the proposed smart UE sorting with power-based sorting are shown in Fig. 4 for QuaDRiGa channel of 48 $\times$ 64 MIMO system. Simulation is done with (MMSE-OSIC + K-best ($K = 8$)) detection algorithm and QAM16 modulation. Power-based sorting considers only power ratios between UEs and ignores correlation, i.e. in the first iteration, the most powerful UE is detected and UE with the lowest power is detected in the last iteration.

V. Smart SR-K-best

The general idea of SR-K-best with parameters $(K, S, \boldsymbol{p})$ from [1] is to determine the $(K - S)$ child nodes according to a fixed order and then apply distributed sorting S times to find the remaining nodes. The vector $\boldsymbol{p}$ is a vector that shows how to choose the first $(K - S)$ child nodes without sorting. For example, vector $\mathbf{p} = [2\ 1\ 1\ 1\ 1\ 0\ 0\ 0]$ means that the detector selects two best child nodes of the first father node and one best child node from the second, third, fourth and fifth father nodes (6 candidates without sorting). The remaining two ($S = 2$) nodes are selected by using a distributed sorting method (D-K-best algorithm) from [8] over the residual child nodes. In the D-K-best algorithm, to apply the sorting procedure we have to reserve the best residual child node from each father node. In the first father node two best child nodes are already pre-selected without sorting, therefore we reserve the third best child node. Then we reserve the second best child node from the second father node because one best child node is also pre-selected in the second father node without sorting, and so on. On the last stage we reserve the first best child node from the eight's father node since this father node has no any pre-selected child nodes. Finally, in the D-K-best algorithm we have the vector $\boldsymbol{v}= [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ corresponding to the 8 reserved child nodes (1 child node per 1 father node). The remaining two nodes (S=2) are selected by sorting among the 8 reserved child nodes defined by the vector $\boldsymbol{v}$. It is claimed that $(K, S, \boldsymbol{p}) = (8, 2,\ [2\ 1\ 1\ 1\ 1\ 0\ 0\ 0])$ could achieve performance similar to K-best with $K = 8$, but in much less sorting resources.

To optimize the SR-K-best algorithm we applied discrete stochastic optimization of $\boldsymbol{p}$ vector with fixed (K,S) parameters in QuaDRiGa channel of 48 $\times$ 64 MIMO system. In fact, a higher K and S parameters always result in better performance in correlated channel, however, in practice $S = K/4$ is a minimal number of sorted candidates to achieve performance similar to K-best. We fixed $(K, S) = (16, 4)$ as a compromise between performance and sorting complexity, but also considered $(K, S, \boldsymbol{p}) = (8, 2,\ [2\ 1\ 1\ 1\ 1\ 0\ 0\ 0])$ for further simulations. Direct D-K-best is inefficient since taking only 1 child node per 1 father node always results in poor performance in high correlated channels. The proposed SR-K-best algorithm consists of parameters $(K, S, \boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q})$ and simultaneous optimization of all of them takes too long time, therefore, discrete stochastic optimization was implemented in Matlab in 3 steps:

***Initialization* :**

Each father node produces 4 child nodes;

***Step* 1 :**

$\boldsymbol{q} = [13\ 14\ 15\ 16]$;
$\boldsymbol{p} = [a_1\ a_2\ ...\ a_{16}]$; $0 \le a_i \le 3$; $a_{i+1} \le a_i$; $\sum a_i = 12$;
$\boldsymbol{v} = [4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4] - \boldsymbol{p}$;
Task: find $\boldsymbol{p}$ with the best gain in BER=10^{-2};

***Step* 2 :**

$\boldsymbol{q} = [13\ 14\ 15\ 16]$; optimized $\boldsymbol{p}$;
$\boldsymbol{v} = [4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4] - \boldsymbol{p}$;
Task: minimize those vector $\boldsymbol{v}$ elements, which cause the least performance degradation in BER=10^{-2} and repeat the minimization until the least degradation $< 0.1dB$;

***Step* 3 :**

Optimized $\boldsymbol{p}$; optimized $\boldsymbol{v}$;
$\boldsymbol{q} = [b_1\ b_2\ b_3\ b_4]$; $b_i \le 16$; $b_{i+1} > b_i$;
Task: find $\boldsymbol{q}$ with the best gain in BER=10^{-2};

After 3 steps of optimization we achieved the result:
$\boldsymbol{p}= [2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]$;
$\boldsymbol{v}= [2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2]$;
$\boldsymbol{q} = [2\ 4\ 6\ 8]$.

The optimized algorithm structure is shown in Fig. 5 and simulation results of optimized SR-K-best are shown in Fig. 6 and Fig. 7 (smart UEs sorting before QRD is used). The algorithm outperforms K-best ($K = 8$) while requires much less computations, i.e. 8 times sorting in $8 * 4 = 32$ child nodes in K-best instead of 4 times sorting in 22 child nodes in the proposed one. It can be found that the initial SR-K-best structure $(K, S, \boldsymbol{p}) = (8, 2,\ [2\ 1\ 1\ 1\ 1\ 0\ 0\ 0])$ requires less computation, but demonstrates huge losses caused by significant correlation between UEs. From Fig. 6 and Fig. 7 leads that OSIC has no losses to (LRA & OSIC) in high correlated scenario. With smart ordering before and after QRD, the proposed low complexity Massive MIMO detection system has only 1 dB losses to ML solution in 48 $\times$ 64 scenario.

VI. Conclusion

We proposed a sorting solution consisting of a new UEs sorting in preprocessing unit and a new SR-K-best structure. The proposed method has a low complexity and results in only 1 dB losses compare to ML solution without using Lattice Reduction Aided (LRA) pre-processing. All simulation was done in QuaDRiGa NLOS channel with 48 UEs moving with averaged speed of 3 km per hour and 64 RX antennas (48 $\times$ 64 MIMO detection system), results are provided for uncoded QAM16 and QAM64 modulations in comparison with other detection methods.

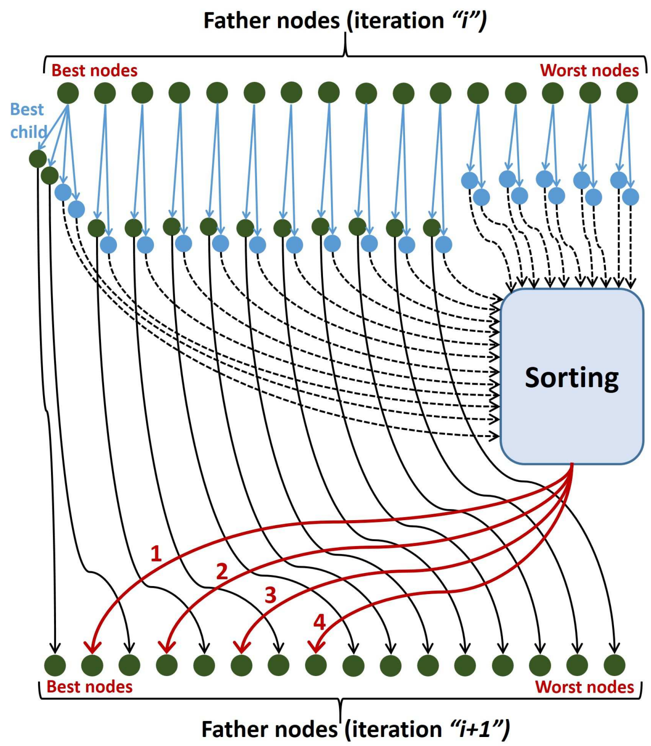

Fig. 5: Implementation structure of the smart SR-K-best

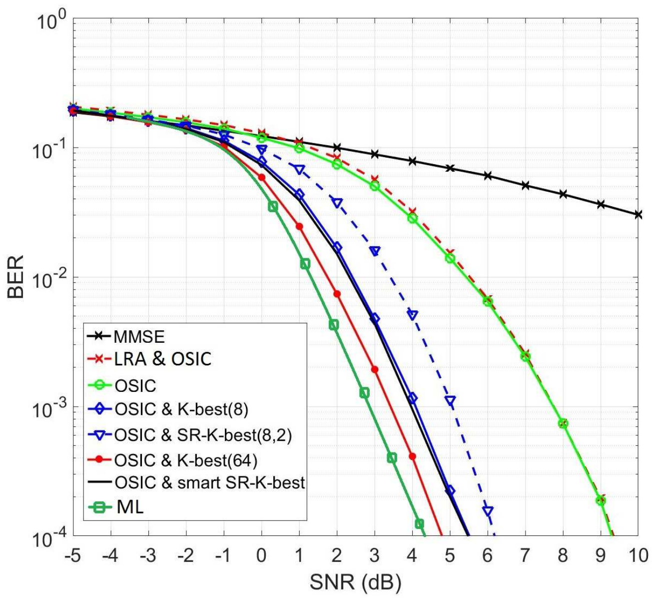

Fig. 6: Simulation results in 48 $\times$ 64 MIMO with QAM16

It is hard to estimate full implementation complexity since "honest" QRD is calculated for pilot symbols only, while the SR-K-best is calculated for each data symbol. Similar situation with UEs sorting before QRD, since its complexity depends on the difference between an ideal ordering in data symbol and power-based ordering in pilot symbol. We can only claim that the proposed algorithm complexity is at least 2

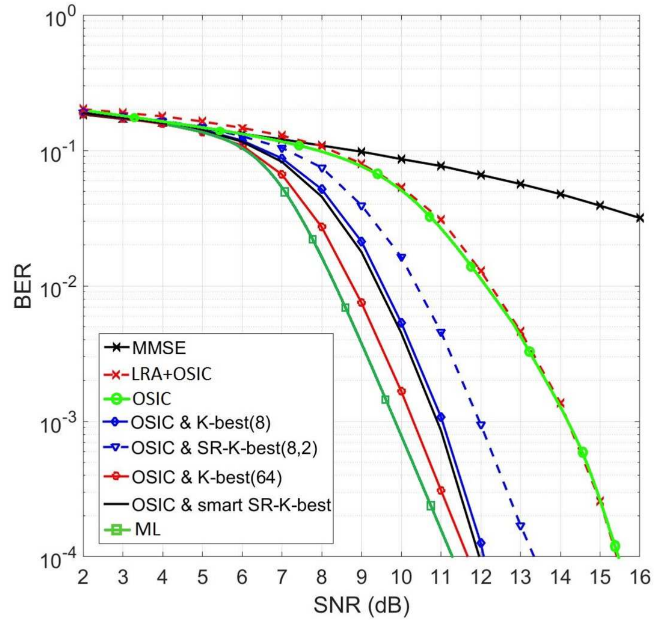

Fig. 7: Simulation results in 48 $\times$ 64 MIMO with QAM64

times lower than the popular LRA-based & K-best solution in the same performance according to our simulations. Moreover, our method is more feasible because of easy LLR calculation, efficient in slicing and has a fixed complexity.

REFERENCES

[1] C. F. Liao, J. Y. Wang, Y. H. Huang, A 3.1 Gb/s 8 x 8 sorting reduced K-best detector with lattice reduction and QR decomposition, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, 2014.

[2] S. Verdu, Multiuser detection, *Cambridge university press*, 1998.

[3] 5G PPP Architecture Working Group, View on 5G Architecture, *Version 2.0*, 2017.

[4] A. Ivanov, S. Kruglik, D. Lakontsev, Cloud MIMO for Smart Parking System, *IEEE 87th Vehicular Technology Conference (VTC-Spring)*, 2018.

[5] M. Li, B. Bougard, E. E. Lopez, A. Bourdoux, D. Novo, L. Van Der Perre, F. Catthoor, Selective spanning with fast enumeration: A near maximum-likelihood MIMO detector designed for parallel programmable baseband architectures, *Proc. IEEE ICC*, 2008.

[6] A. M. Mussi, B. F. Costa, T. Abro, Wireless Pers Commun, *Proc. IEEE ICC*, https://doi.org/10.1007/s11277-016-3807-6, 2017.

[7] Z. Guo, P. Nilsson, A VLSI implementation of MIMO detection for future wireless communications, *Proc. IEEE Pers. Indoor Mobile Radio Commun.*, pp. 2852-2856, 2003.

[8] M. Shabany, P. G. Gulak, The application of lattice-reduction to the K-best algorithm for near-optimal MIMO detection, *IEEE International Symposium on Circuits and Systems*, 2008.

[9] P. L. Chiu, L. Z. Huang, Y. H. Huang, Scalable interpolation-based QRD architecture for subcarrier-grouped-ordering MIMO-OFDM system, *Proc. 43th IEEE Asilomar Conf. on Signals Systems and Computers*, pp. 708-712, 2009.

[10] Ramanathan, R. & Jayakumar, A support vector regression approach to detection in large-MIMO systems, *M. Telecommun Syst, https://doi.org/10.1007/s11235-016-0202-2*, 2017.

[11] N. Samuel, T. Diskin, A. Wiesel, Deep MIMO detection, *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2017.

[12] T. J. O'Shea, J. Hoydis, An introduction to deep learning for the physical layer, *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, 2017.

[13] http://quadriga-channel-model.de/

[14] P.-L. Chiu, L.-Z. Huang, Li-Wei Chai, Yuan-Hao Huang, Interpolation-Based QR Decomposition and Channel Estimation Processor for MIMO-OFDM System, *IEEE Trans. Circuits and Systems*, vo1. 58, 2011.