An Unconstrained Likelihood Ascent Based Detection Algorithm for Large MIMO Systems

Abhay Kumar Sah, *Student Member, IEEE*, and A. K. Chaturvedi, *Senior Member, IEEE*

***Abstract*— Neighborhood search algorithms have been proposed for detection in large multiple-input multiple-output systems. They iteratively search for the best vector in a fixed neighborhood. A better way could be to look for an update which is not restricted to a fixed neighborhood. Motivated by this, we formulate a problem to maximize the reduction in maximum likelihood (ML) cost and use it to derive an expression for updating the current solution. Using this update and a likelihood function regarding the locations of errors, we propose an unconstrained likelihood ascent search (ULAS) algorithm. ULAS seeks to provide the maximum reduction in ML cost by finding an update which is not restricted to be in a fixed neighborhood. Using simulations, the proposed algorithm has been shown to provide better error performance for uncoded systems than existing algorithms, at lower complexity. We also show that ULAS is amenable to lattice reduction, which helps in obtaining two variants leading to further improvements in performance.**

***Index Terms*— Large MIMO, neighborhood search algorithms, LAS algorithms, lattice reduction techniques.**

I. Introduction

DRIVEN by the need for high data rate for future wireless standards [1], [2], the number of antenna pairs in multiple-input multiple-output (MIMO) systems may be scaled up from an order of ten to hundred [3], [4]. Such systems are known as large MIMO systems. The realization of such systems requires several things, one of them being the availability of reliable and computationally efficient detectors. This is because the performance of low complexity detectors such as zero forcing (ZF) and minimum mean square error (MMSE) [5], [6] deteriorates with increasing number of antennas while the complexity of near maximum likelihood (ML) detectors like the ones reported in [7]–[13] increases rapidly with the number of antennas.

Several algorithms have been reported in the literature to address the issue of large MIMO detection. These can be broadly classified as neighborhood search algorithms [14]–[20], graph based algorithms [21]–[23], sparsity boosted algorithms [24]–[26], lattice reduction (LR) based algorithms [27]–[29], and other algorithms like [30]–[32]. Among these, neighborhood search algorithms have been well investigated and have also been used to overcome the limitations of other algorithms. For example, the applicability of a Markov random field (MRF) based graphical model or a factor graph (FG) approach can be enhanced to higher order constellations using a neighborhood search algorithm [22]. Similarly, [33] and [34] combine neighborhood search algorithms with sparsity and LR based techniques, respectively.

In the class of neighborhood search algorithms, likelihood ascent search (LAS) algorithm [14] is the first reported algorithm. This algorithm starts with an initial guess and searches for the best solution in the neighborhood (including the initial guess) until the improvement in ML cost saturates. The complex constellation version of this scheme was given in [15]. Two variants of LAS algorithm, namely, one symbol LAS (1-LAS) and multistage LAS (MLAS) were proposed in [15]. The 1-LAS algorithm provides a low complexity solution while MLAS requires significantly higher computations but provides improved error performance. To improve the performance in terms of error rate and complexity, [16] suggested some policies to avoid the early termination problem in 1-LAS and named it as reactive tabu search (RTS) while [17] suggested initialization of 1-LAS with multiple initial vectors. However, the improvement in performance comes at the price of extra computations and also the improvement is limited to small constellations only. To improve the performance for large constellations, layered tabu search (LTS) was proposed in [19], which employs the RTS algorithm multiple times, requiring considerably higher computations.

It may be noted that all of the above algorithms search in a fixed neighborhood. However, the idea of searching in a fixed neighborhood has limitations. Thus, the ML solution may not lie in the searched space and even if it lies, the search process may take a large number of intermediate vectors to converge. Instead of searching in a fixed neighborhood of the current solution vector, a better way will be updating it with no constraints on the neighborhood.

Motivated by this, we propose an algorithm in which the solution vector for the next iteration is a sum of the solution vector of the current iteration and an update vector. The algorithm essentially consists of two parts. In the first part, we compute a likelihood of a given location in the solution vector being in error. The likelihoods are arranged in the decreasing order, in terms of their magnitude. The locations corresponding to the first few, say L, elements are selected. The second part consists of an expression for update assuming these L locations to be in error. We compute the update for all values of L and select the one, the solution vector corresponding to which, provides the minimum ML cost. The algorithm stops

Manuscript received April 16, 2016; revised September 13, 2016 and January 19, 2017; accepted January 23, 2017. Date of publication March 13, 2017; date of current version April 7, 2017. This work was supported by Tata Consultancy Services Ltd. Project TCS/CS/20110191. The associate editor coordinating the review of this paper and approving it for publication was C. Tepedelenlioglu.

A. K. Sah is with the Department of Electrical Engineering, IIT Kanpur, Kanpur 208016, India (e-mail: abhaysah@iitk.ac.in).

A. K. Chaturvedi is with the Department of Electrical Engineering, IIT Kanpur, Kanpur 208016, India and also with the Department of Electronics and Communication Engineering, IIT Roorkee, Roorkee 247667, India (e-mail: akc@iitk.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2017.2661283

when there is no further reduction in the ML cost. We call the proposed algorithm as the unconstrained LAS (ULAS) algorithm. Simulation results show that compared to existing algorithms ULAS has better error performance, that too at a lower complexity.

Further, we derive a lattice reduction (LR) aided extension of the ULAS algorithm. For this, we customize the existing element based LR (ELR) algorithm [27] so that it fits in the ULAS algorithm. This leads to two variants of ELR, namely, modified primal ELR (MPELR) and modified dual ELR (MDELR). Simulation results show that the LR aided algorithms have much improved performance, especially for high order constellations at the cost of some additional complexity compared to ULAS algorithm.

The rest of the paper is structured as follows: Section II describes the system model while existing neighborhood search algorithms and their scope for improvement is discussed in Section III. We formulate an optimization problem in Section IV and propose a detection algorithm in Section V. In Section VI, we analyze its complexity. Section VII generalizes the results to facilitate the application of LR techniques and two variants of LR techniques have been proposed in Section VIII. Section IX provides the simulation results and finally, Section X concludes the paper. In this paper, we use boldface capital letters to denote matrices, boldface small letters to denote vectors, small letters to denote elements of a vector/matrix, and calligraphic letters to denote a set. The notations $\bar{(\cdot)}$, $(\cdot)^T$, $\mathbf{I}_N$, and $\mathbb{Z}$ stand for complex quantities, transpose, an $N \times N$ identity matrix, and an integer set, respectively. Similarly, $\Re\{\cdot\}$ represents the real part of a complex number and $\Im\{\cdot\}$ represents the imaginary part of a complex number.

II. System Model

Consider a MIMO system using N_t transmit antennas for transmission and N_r receive antennas for reception ($N_t \leq N_r$). The input-output relationship of the system can be mathematically modeled as

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

where $\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{N_r})^T$ in which $\bar{y}_i$ represents data received at the ith receive antenna and $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{N_t})^T$ is the transmitted signal vector where $\bar{x}_i \in \bar{\Omega}$ represents data transmitted through the ith transmit antenna. Here $\bar{\Omega}$ is a set of M complex symbols taken from a square constellation. The channel matrix $\bar{\mathbf{H}}$ is of dimension $N_r \times N_t$ with each coefficient $\bar{h}_{ij} \sim \mathcal{CN}(0, 1)$ and $\bar{\mathbf{n}} = (\bar{n}_1, \bar{n}_2, \ldots, \bar{n}_{N_r})^T$ represents an $N_r \times 1$ i.i.d. additive white Gaussian noise (AWGN) vector with each $\bar{n}_i \sim \mathcal{CN}(0, \sigma^2)$. The complex system model in (1) can be formulated as an equivalent real system model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2}$$

where $\mathbf{y} = (\Re\{\bar{\mathbf{y}}\}^T \ \Im\{\bar{\mathbf{y}}\}^T)^T$ is a $2N_r \times 1$ real equivalent received vector and $\mathbf{x} = (\Re\{\bar{\mathbf{x}}\}^T \ \Im\{\bar{\mathbf{x}}\}^T)^T$ is a $2N_t \times 1$ real equivalent transmit vector where each $x_i \in \Omega$. Now, the set $\Omega = \{\pm 1, \pm 3, \ldots, \pm(\sqrt{M} - 1)\}$ is a set of $\sqrt{M}$ real symbols drawn from a one dimensional constellation, $\mathbf{n} = (\Re\{\bar{\mathbf{n}}\}^T \ \Im\{\bar{\mathbf{n}}\}^T)^T$ is a $2N_r \times 1$ equivalent noise vector and $\mathbf{H}$ is the $2N_r \times 2N_t$ equivalent channel matrix given by

$$\mathbf{H} = \begin{bmatrix} \Re\{\bar{\mathbf{H}}\} & -\Im\{\bar{\mathbf{H}}\} \\ \Im\{\bar{\mathbf{H}}\} & \Re\{\bar{\mathbf{H}}\} \end{bmatrix}. \tag{3}$$

We will use this real system model (2) throughout the paper.

At the receiver our objective is to find the vector $\mathbf{x}$ among all the possible $\sqrt{M}^{2N_t}$ transmit vectors which is nearest to the received signal vector $\mathbf{y}$ for the given channel matrix $\mathbf{H}$. Mathematically, this is stated as

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \Omega^{2N_t}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{4}$$

and is known as the ML solution. Here $\|\cdot\|$ denotes L_2 norm. We define the Euclidean cost function as

$$\phi(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \sum_{i=1}^{2N_r} \left| y_i - \textstyle\sum_{j=1}^{2N_t} h_{ij} x_j \right|^2, \tag{5}$$

and call it ML cost, or simply cost, in the rest of the paper. The computational complexity of ML detection is exponential in nature and cannot be applied practically even for a small number of antenna pairs. In the next section, we discuss several neighborhood search algorithms which have been suggested for large MIMO systems.

III. Scope for Improvement in Neighborhood Search Algorithms

The basic idea of neighborhood search algorithms is to initialize with an initial solution vector and then search for the best solution in the neighborhood of the initial solution vector. The process is repeated as long as there is a reduction in the ML cost (5).

A. Definition of Neighborhood

The L symbol neighborhood $\mathcal{N}_L(\mathbf{x})$ of a vector $\mathbf{x}$ is defined as follows: Consider a vector $\tilde{\mathbf{x}}$ which differs from $\mathbf{x}$ at exactly L symbols then $\tilde{\mathbf{x}} \in \mathcal{N}_L(\mathbf{x})$. There will be a total of $\binom{2N_t}{L}$ such $\tilde{\mathbf{x}}$. The sets of indices at which $\mathbf{x}$ and $\tilde{\mathbf{x}}$ differ are denoted by I_k, where $k = 1, 2, \ldots, \binom{2N_t}{L}$. Mathematically, a neighboring vector $\tilde{\mathbf{x}} \in \mathcal{N}_L(\mathbf{x})$ can be expressed as

$$\tilde{x}_i = \begin{cases} \omega_j, & i \in I_k, \ \omega_j \neq x_i, \ \& \, \omega_j \in \Omega \\ x_i, & i \notin I_k, \end{cases} \tag{6}$$

where $i = 1, 2, \ldots, 2N_t$ and $j = 1, 2, \ldots, \sqrt{M}$. Thus, there are $(\sqrt{M} - 1)^L \binom{2N_t}{L}$ vectors in $\mathcal{N}_L(\mathbf{x})$. For example, let us consider the set $\Omega = \{-3, -1, 1, 3\}$ and a vector $\mathbf{x} = [1 \ 3 \ -1]$ of length 3. The 1-symbol neighborhood of $\mathbf{x}$ is constructed by flipping each element of $\mathbf{x}$ with the elements of the set Ω, one at a time. Thus, corresponding to the first element of $\mathbf{x}$, there are three neighboring vectors: $[-3 \ 3 \ -1]$, $[-1 \ 3 \ -1]$ and $[3 \ 3 \ -1]$ in the 1-symbol neighborhood of $\mathbf{x} = [1 \ 3 \ -1]$. Similarly, for the other two locations, we have six more vectors.

B. Scope for Improvement

Based on the above neighborhood definition, there exist several neighborhood search algorithms such as 1-LAS algorithm [14], MLAS algorithm [15], multiple output selection LAS [17], RTS [16], random restart reactive tabu search [18], LTS [19], etc. Here, we briefly describe the limitations of these algorithms, and then explore the scope for improvement in neighborhood search algorithms.

- 1-LAS is a relatively simple, low complexity algorithm, which searches for the best vector in a one symbol neighborhood. However, it can get trapped in a local minima, resulting in a limited space search. This leads to a loss in error performance.
- MLAS proposes a policy to escape from local minima. Thus, whenever 1-LAS is trapped in a local minima, we search for the best vector in two and three symbol neighborhoods leading to significantly higher computations.
- To reduce the complexity of MLAS and to find a better solution than 1-LAS, the RTS algorithm keeps track of the moves and tries to avoid any trap. However, it leads to a higher computational complexity (compared to 1-LAS) and also the error performance deteriorates with the constellation size.
- LTS algorithm has been proposed to improve the performance for larger constellations. The complexity of LTS increases exponentially with the number of transmit antennas.

Based on the above observations, it can be seen that except MLAS, all of these algorithms limit the search to a one symbol neighborhood, or in other words update only one element at each iteration. This may lead to either early termination, as in the case of 1-LAS, or require a higher number of iterations as in the case of RTS and LTS, leading to a higher computational complexity. Motivated by this, in the next section we revisit the neighborhood search algorithm and construct a new optimization problem.

IV. Problem Formulation

Let us denote the solution vector at the start of rth iteration as $\mathbf{x}^{(r)}$ and the solution vector at the end of rth iteration i.e. at the start of $(r+1)$th iteration as $\mathbf{x}^{(r+1)}$. These two solution vectors are related as $\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \mathbf{p}^{(r)}$ where $\mathbf{p}^{(r)}$ is the update vector. The ML cost of $\mathbf{x}^{(r+1)}$ can be expressed as a function of the cost at the previous iteration and the update vector, using Taylor series expansion, as

$$\begin{aligned}\phi(\mathbf{x}^{(r+1)}) &= \phi(\mathbf{x}^{(r)} + \mathbf{p}^{(r)}) \\ &= \phi(\mathbf{x}^{(r)}) + \phi'(\mathbf{x}^{(r)})\mathbf{p}^{(r)} + \frac{1}{2}\mathbf{p}^{(r)^T}\phi''(\mathbf{x}^{(r)})\mathbf{p}^{(r)}, \quad (7)\end{aligned}$$

where

$$\phi'(\mathbf{x}^{(r)}) = -2(\mathbf{y} - \mathbf{H}\mathbf{x}^{(r)})^T\mathbf{H}, \quad (8)$$

$$\phi''(\mathbf{x}^{(r)}) = 2\mathbf{H}^T\mathbf{H}. \quad (9)$$

Since the cost function is quadratic in nature, the terms corresponding to the third and higher derivatives in the Taylor series expansion in (7), will be zero. The difference in the cost for two consecutive iterations is given by

$$\begin{aligned}\Delta\phi(\mathbf{p}^{(r)}) &= \phi(\mathbf{x}^{(r+1)}) - \phi(\mathbf{x}^{(r)}) \\ &= -2(\mathbf{y} - \mathbf{H}\mathbf{x}^{(r)})^T\mathbf{H}\mathbf{p}^{(r)} + \mathbf{p}^{(r)^T}\mathbf{H}^T\mathbf{H}\mathbf{p}^{(r)} \\ &= \mathbf{p}^{(r)^T}\mathbf{W}\mathbf{p}^{(r)} - 2\,\mathbf{f}^{(r)}\mathbf{p}^{(r)}, \quad (10)\end{aligned}$$

where $\mathbf{W} \triangleq \mathbf{H}^T\mathbf{H}$ and $\mathbf{f}^{(r)} \triangleq (\mathbf{y} - \mathbf{H}\mathbf{x}^{(r)})^T\mathbf{H}$.

A. Unconstrained Optimization Problem

Usually, a move from the rth iteration to the $(r+1)$th iteration is permitted if and only if it reduces the ML cost, i.e. $\Delta\phi < 0$. For a given rth iteration, the update which minimizes $\Delta\phi(\mathbf{p}^{(r)})$ or maximizes the reduction in ML cost is the optimal update $\mathbf{p}^o$. We wish to explore the optimal update which is not constrained to belong to a specified neighborhood, i.e. it can be from anywhere in the entire space. We refer to it as an unconstrained update which can be mathematically expressed as

$$\begin{aligned}\mathbf{p}^o &= \underset{\mathbf{p}^{(r)} \in \vartheta^{2N_t}}{\operatorname{argmin}} \ \Delta\phi(\mathbf{p}^{(r)}) \\ &= \underset{\mathbf{p}^{(r)} \in \vartheta^{2N_t}}{\operatorname{argmin}} \ \mathbf{p}^{(r)^T}\mathbf{W}\mathbf{p}^{(r)} - 2\,\mathbf{f}^{(r)}\mathbf{p}^{(r)}, \quad (11)\end{aligned}$$

where $\vartheta = \{\omega_i - \omega_j | \forall \omega_i, \omega_j \in \Omega\}$. This unconstrained optimization problem will achieve the ML solution. It is NP-hard and requires very high computational resources. The elements of $\mathbf{p}^{(r)}$ can take values from the set ϑ, the size of which is $2\sqrt{M} - 1$. Hence there are $(2\sqrt{M} - 1)^{2N_t}$ possible updates. Therefore, finding an optimal update is computationally expensive.

B. Optimization Problem Revisited

Let us revisit the problem stated in (11) and expand the objective function as follows

$$\Delta\phi = \sum_{i=1}^{2N_t}\sum_{j=1}^{2N_t}\langle \mathbf{h}_i, \mathbf{h}_j\rangle p_i^{(r)} p_j^{(r)} - 2\sum_{i=1}^{2N_t} f_i^{(r)} p_i^{(r)}, \quad (12)$$

where $p_i^{(r)}$ denotes the ith element of $\mathbf{p}^{(r)}$, $\mathbf{h}_i$ represents the ith column of the channel matrix $\mathbf{H}$, and $\langle \mathbf{a}, \mathbf{b}\rangle$ denotes the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. In the above expression, any one $p_i^{(r)} = 0$ will lead to $4N_t$ terms to zero and the dimension of the problem will reduce by one. Thus, if we need to update L symbols, there will be $(2N_t - L)$ zeros in $\mathbf{p}^{(r)}$ leading to reduction in the dimension of the problem in (11). For a given value of L, there are $\binom{2N_t}{L}$ I_k's where I_k is as defined in Section III-A. For a given L and I_k, we can express (11) as

$$\mathbf{p}_{(L,I_k)}^{(r)o} = \underset{\mathbf{p}_{(L,I_k)}^{(r)} \in \vartheta^L}{\operatorname{argmin}} \ \mathbf{p}_{(L,I_k)}^{(r)^T}\mathbf{W}_{(L,I_k)}\mathbf{p}_{(L,I_k)}^{(r)} - 2\,\mathbf{f}_{(L,I_k)}^{(r)}\mathbf{p}_{(L,I_k)}^{(r)}, \quad (13)$$

where $\mathbf{p}_{(L,I_k)}^{(r)}$ is an $L \times 1$ update vector and $\mathbf{W}_{(L,I_k)}$ is an $L \times L$ sub matrix constructed by deleting the $(2N_t - L)$ rows and $(2N_t - L)$ columns of $\mathbf{W}$, the indices of which are not part of I_k. Similarly $\mathbf{f}_{(L,I_k)}^{(r)}$ is an $L \times 1$ vector, formed by deleting the elements from $\mathbf{f}^{(r)}$, the indices of which are not part of I_k.

The update which minimizes (13) is the best update at the rth iteration for the given L and I_k and we denote it by $\mathbf{p}^{(r)^o}_{(L,I_k)}$.

Finding this update needs an exhaustive search over $(2\sqrt{M}-1)^L$ possibilities which is computationally expensive for large L. Therefore, we take an alternate approach. Assuming the variable $\mathbf{p}^{(r)}_{(L,I_k)}$ to be continuous, we check the convexity of the problem. It is easy to verify that the Hessian of the objective function in (13) is positive semi-definite i.e. $\nabla^2\Delta\phi \succeq 0$. Taking its partial derivative and equating it to zero, we get

$$\nabla\Delta\phi = 2\mathbf{W}_{(L,I_k)}\mathbf{p}^{(r)}_{(L,I_k)} - 2\,\mathbf{f}^{(r)^T}_{(L,I_k)} = 0. \tag{14}$$

Since the solution obtained from (14) need not belong to the alphabet set ϑ, we approximate the solution to the nearest alphabet in the set ϑ. Thus, the update for a given L and I_k will be given by

$$\mathbf{p}^{(r)^*}_{(L,I_k)} = \left\lceil \mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} \right\rfloor_{\vartheta}, \tag{15}$$

where $\lceil\cdot\rfloor_{\vartheta}$ represents element-wise rounding on the alphabet set ϑ. The update in (15) is easy to compute and can be expected to give a reasonably good performance. In the sequel, we refer to it as a reasonably good (RG) update. From the above it is clear that the RG update is not necessarily the optimal update. Interestingly, the optimal one symbol update derived in [15] is a special case of (15) for $L = 1$.

Let us denote the update as

$$\mathbf{p}^{(r)^*}_{(L,I_k)} = \mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)}_{(L,I_k)}, \tag{16}$$

where $\boldsymbol{\delta}^{(r)}_{(L,I_k)}$ is the rounding error. Substituting the above solution in (13), we get the expression for the reduction in the cost i.e. $\Delta\phi^{(r)}_{(L,I_k)}$ as given in (17), shown at the bottom of the page. The values of L and I_k for which the reduction in the cost function is maximized i.e. $\Delta\phi^{(r)}_{(L,I_k)}$ is minimized can be used to find the best RG update $\mathbf{p}^*$. We express this equivalent optimization problem in (18), shown at the bottom of the page, and denote the value of L and I_k corresponding to $\mathbf{p}^*$ as L^* and I_k^*, respectively.

Based on this formulation, we can obtain a sub-optimal solution of (11) by solving the following two problems: (i) Find the (L^*, I_k^*) pair and (ii) Determine the RG update among all the $(2\sqrt{M}-1)^L$ possibilities. The solution for the second problem can be easily obtained using (15) but the first problem corresponds to the optimization problem in (18). It may be noted that finding the best (L^*, I_k^*) leads to a search over $\sum_{L=1}^{2N_t}\binom{2N_t}{L} = (2^{2N_t}-1)$ updates. Although it is less than the number of computations in (11), it is still computationally expensive. We address this issue in the next section by using an approximation of (18) to cut down the number of candidate updates.

V. Proposed Unconstrained LAS Algorithm

In this section, we combine the ideas of the previous section to propose an unconstrained LAS algorithm which we refer to as the ULAS algorithm. A stepwise description of the ULAS algorithm follows.

A. Initialization

The algorithm is initialized with a solution vector $\mathbf{x}^{(0)}$. We may start with a random guess, but just like other neighborhood search algorithms, the error performance and complexity will depend on the accuracy of the initial solution. Hence, we choose to initialize with either a matched filter (MF), a zero forcing (ZF) or a minimum mean square error (MMSE) solution, the expressions for which are

$$\begin{aligned}
\mathbf{x}^{(0)}_{MF} &= \left\lceil \mathbf{H}^T\mathbf{y} \right\rfloor_{\Omega},\\
\mathbf{x}^{(0)}_{ZF} &= \left\lceil (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \right\rfloor_{\Omega},\\
\mathbf{x}^{(0)}_{MMSE} &= \left\lceil (\mathbf{H}^T\mathbf{H} + \sigma^2\mathbf{I}_{N_t})^{-1}\mathbf{H}^T\mathbf{y} \right\rfloor_{\Omega},
\end{aligned} \tag{19}$$

where $\lceil\cdot\rfloor_{\Omega}$ represents element-wise rounding to the set Ω and the subscripts denote the type of the receiver. One can begin with any one of the above initial vectors.

B. Computation of $\mathbf{f}^{(r)}$ and $\mathbf{W}$

We compute $\mathbf{W}$ and $\mathbf{f}^{(r)}$ using the relationships: $\mathbf{W} = \mathbf{H}^T\mathbf{H}$ and $\mathbf{f}^{(r)} = (\mathbf{y} - \mathbf{H}\mathbf{x}^{(r)})^T\mathbf{H}$. It may be noted that $\mathbf{W}$ needs to be computed only once while $\mathbf{f}^{(r)}$ needs to be computed at each iteration.

C. Determining the Candidate Sequence $\mathcal{T}$

For a given $\mathbf{f}$ and $\mathbf{W}$, the optimization problem in (18) is a function of L and I_k. We would like to determine the L and I_k which will cause the maximum reduction in the ML cost. Since finding an optimal L and I_k is computationally expensive, we invoke an approximation using a property of large MIMO systems. We know that because of the channel-hardening behavior [35], the channel matrix $\mathbf{H}$ in a large MIMO system is quasi-orthogonal [36]. Therefore, as discussed in [1] $\mathbf{W}$ and $\mathbf{W}^{-1}$ will be asymptotically diagonal matrices, and hence can be taken to be well-conditioned. Further, from (16) we note that the vector $\boldsymbol{\delta}^{(r)}_{(L,I_k)}$ has its origin in the operation $\mathbf{H}^{-1}\mathbf{n}$. This vector is the same as the noise amplification vector term obtained in linear detectors, the components of which are known to be low when the channel matrix is well conditioned [37]. Since the matrices involved herein are also well conditioned, the components in $\boldsymbol{\delta}^{(r)}_{(L,I_k)}$ can be taken to be low. Therefore the second term in (18) is

$$\begin{aligned}
\Delta\phi^{(r)}_{(L,I_k)} &= \left(\mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)}_{(L,I_k)}\right)^T\mathbf{W}_{(L,I_k)}\left(\mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)}_{(L,I_k)}\right) - 2\,\mathbf{f}^{(r)}\left(\mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)}_{(L,I_k)}\right)\\
&= -\mathbf{f}^{(r)}_{(L,I_k)}\mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)^T}_{(L,I_k)}\mathbf{W}_{(L,I_k)}\boldsymbol{\delta}^{(r)}_{(L,I_k)},
\end{aligned} \tag{17}$$

$$L^*, I_k^* = \underset{L,I_k}{\operatorname{argmin}}\left\{-\mathbf{f}^{(r)}_{(L,I_k)}\mathbf{W}^{-1}_{(L,I_k)}\mathbf{f}^{(r)^T}_{(L,I_k)} + \boldsymbol{\delta}^{(r)^T}_{(L,I_k)}\mathbf{W}_{(L,I_k)}\boldsymbol{\delta}^{(r)}_{(L,I_k)}\right\}. \tag{18}$$

likely to be small compared to the first term, thus justifying the following approximation

$$\underset{L,I_k}{\operatorname{argmin}} \sum_{i\in I_k} -\frac{f_i^{(r)^2}}{w_{i,i}} = \underset{L,I_k}{\operatorname{argmax}} \sum_{i\in I_k} \frac{f_i^{(r)^2}}{w_{i,i}}. \tag{20}$$

We propose to use $\frac{f_i^{(r)^2}}{w_{i,i}}$ as a likelihood of the erroneous locations ($i = 1, 2, \ldots, 2N_t$). We sort the indices in the descending order of this likelihood and denote the resulting sequence by $\mathcal{T}$. Using this candidate sequence it will be easy to compute L^* and I_k^*. It may be noted that the proposed likelihood is same as in [38] although its manner of usage is different here, as discussed next.

D. Arranging $\mathbf{x}^{(r)}$, $\mathbf{f}^{(r)}$, and $\mathbf{W}$

Now we use the order in $\mathcal{T}$ to arrange the elements in $\mathbf{x}^{(r)}$. This means the symbol which comes earlier is more likely to be in error than the symbol which comes later. Thus an RG update using (15) on the first L elements of $\mathbf{x}^{(r)}$ can be used to arrive at the next solution vector, in an L symbol neighborhood search. Since the expression for update is a function of $\mathbf{f}^{(r)}$ and $\mathbf{W}$, we need to do the following: (i) Arrange the elements of $\mathbf{f}^{(r)}$ in the same order as $\mathcal{T}$, (ii) Arrange the columns of $\mathbf{W}$ in the same order as $\mathcal{T}$ and lastly, (iii) The rows of $\mathbf{W}$ obtained in (ii) are arranged in the same order as $\mathcal{T}$. We will use these arranged $\mathbf{x}^{(r)}$, $\mathbf{f}^{(r)}$, and $\mathbf{W}$ in the next subsection.

E. RG Update $\mathbf{p}_{(L,I_k)}^{(r)^*}$ and Rounding Error $\boldsymbol{\delta}_{(L,I_k)}^{(r)}$

For a given value of L, we select the first L elements of $\mathcal{T}$, $\mathbf{x}^{(r)}$, and $\mathbf{f}^{(r)}$ and denote it as I_k, $\mathbf{x}_{(L,I_k)}^{(r)}$, and $\mathbf{f}_{(L,I_k)}^{(r)}$, respectively. Similarly, we select the $L \times L$ matrix formed by the intersection of the first L rows and the first L columns of $\mathbf{W}$ and denote it as $\mathbf{W}_{(L,I_k)}$. Now we compute the RG update $\mathbf{p}_{(L,I_k)}^{(r)^*}$ for a given (L, I_k) pair using (15). Using this update a candidate $(r+1)$th solution vector is given by

$$\mathbf{x}_{(L,I_k)}^{(r+1)} = \left\lceil \mathbf{x}_{(L,I_k)}^{(r)} + \mathbf{p}_{(L,I_k)}^{(r)^*} \right\rfloor_{\Omega}, \tag{21}$$

where $\lceil \cdot \rfloor_{\Omega}$ denotes rounding to the set Ω. This is required because some of the elements of the $(r+1)$th solution vector may lie outside the set Ω. The corresponding rounding error $\boldsymbol{\delta}_{(L,I_k)}^{(r)}$ is given by

$$\boldsymbol{\delta}_{(L,I_k)}^{(r)} = \mathbf{x}_{(L,I_k)}^{(r+1)} - \mathbf{x}_{(L,I_k)}^{(r)} - \mathbf{p}_{(L,I_k)}^{(r)}. \tag{22}$$

Since L can take values from 1 to $2N_t$, we get $2N_t$ candidate solution vectors for the $(r+1)$th iteration.

F. Update for the rth Iteration and Terminaton

Once we have $2N_t$ possible updates, we need to choose the best one for the $(r+1)$th iteration. By calculating the update for all possible values of L, we are effectively computing the RG update for all possible sizes of the neighborhood. We select the RG update $\mathbf{p}_{(L,I_k)}^{(r)^*}$ for which $\Delta\phi_{(L,I_k)}^{(r)}$ (17) is minimum and denote the corresponding (L, I_k) as (L^*, I_k^*). Since the update vector in Section V-E has L^* elements, we need to append $(2N_t - L^*)$ zeros to $\mathbf{p}_{(L^*,I_k^*)}^{(r)^*}$ and then reorder the elements according to $\mathcal{T}$. This rearranged version of $\mathbf{p}_{(L^*,I_k^*)}^{(r)^*}$ is the best RG update for the rth iteration and we denote it by $\mathbf{p}^{(r)^*}$. This is added to the unsorted $\mathbf{x}^{(r)}$ (i.e. $\mathbf{x}^{(r)}$ before Section V-D) and rounded off to the set Ω to arrive at the solution vector for the $(r+1)$th iteration.

Algorithm 1 ULAS Algorithm

Input : $\mathbf{y}$, $\mathbf{H}$, Ω
Output: $\hat{\mathbf{x}}$

Initialization $r = 0$;
$\mathbf{x}^r \leftarrow$ output of MF/ZF/MMSE detector;
$Cost_{next} \leftarrow \|\mathbf{y} - \mathbf{H}\mathbf{x}^r\|^2$ & $Cost_{pre} \leftarrow \infty$;
while $Cost_{next} < Cost_{pre}$ **do**
  Compute $\mathbf{f}^{(r)} = (\mathbf{y} - \mathbf{H}\mathbf{x}^r)^T\mathbf{H}$ and $\mathbf{W} = \mathbf{H}^T\mathbf{H}$;
  $\mathcal{T} \leftarrow$ sort the indices according to $\frac{f_i^{(r)^2}}{w_{i,i}}$ $\forall i = 1, 2, \ldots, 2N_t$ in descending order;
  $Cost_{pre} \leftarrow Cost_{next}$;
  $\mathbf{x}^{temp} \leftarrow \mathbf{x}^r$;
  Arrange $\mathbf{x}_{temp}$, $\mathbf{f}^{(r)}$, and $\mathbf{W}$ according to $\mathcal{T}$;
  for $L \leftarrow 1$ **to** $length(\mathcal{T})$ **do**
    $\mathbf{W}_L = \mathbf{W}_{1:L,1:L}$;
    $\mathbf{f}_L^{(r)} = \mathbf{f}_{(1:L)}^{(r)}$, $\mathbf{x}_L^{(r)} = \mathbf{x}_{(1:L)}^{temp}$;
    if $j = 1$ **then**
      $\mathbf{W}_1^{-1} = \frac{1}{\mathbf{W}_{1,1}}$;
    else
      $\mathbf{b} = \mathbf{w}_{1:L-1,L}$;
      $\rho = w_{L,L} - \mathbf{b}^T\mathbf{W}_{L-1}^{-1}\mathbf{b}$;
      $\mathbf{W}_L^{-1} = \begin{bmatrix} \mathbf{W}_{L-1}^{-1} + \frac{1}{\rho}\mathbf{W}_{L-1}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{W}_{L-1}^{-1} & -\frac{1}{\rho}\mathbf{W}_{L-1}^{-1}\mathbf{b} \\ -\frac{1}{\rho}\mathbf{b}^T\mathbf{W}_{L-1}^{-1} & \frac{1}{\rho} \end{bmatrix}$;
    $\mathbf{p}_L^{(r)} = \mathbf{W}_L^{-1}\mathbf{f}_L^{(r)}$ and $\mathbf{p}_L^{(r)^*} = \left\lceil \mathbf{p}_L^{(r)} \right\rfloor_{\vartheta}$;
    $\widetilde{\mathbf{x}}_L^{(r)} = \left\lceil \mathbf{x}_L^{(r)} + \mathbf{p}_L^{(r)^*} \right\rfloor_{\Omega}$;
    $\boldsymbol{\delta}_L^{(r)} = \widetilde{\mathbf{x}}_L^{(r)} - \mathbf{p}_L^{(r)}$;
    $Cost_{temp} = -\mathbf{f}_L^{(r)^T}\mathbf{W}_L^{-1}\mathbf{f}_L^{(r)} + \boldsymbol{\delta}_L^{(r)^T}\mathbf{W}_L\boldsymbol{\delta}_L^{(r)}$;
    if $(Cost_{temp} < Cost_{next})$ **then**
      $\mathbf{x}^{temp} \leftarrow \widetilde{\mathbf{x}}_L^{(r)}$, $L^* \leftarrow L$ and $Cost_{next} \leftarrow Cost_{temp}$;
  Add $(2N_t - L^*)$ zeros in $\mathbf{x}^{temp}$ and rearrange it according to $\mathcal{T}$;
  $\mathbf{x}^{r+1} \leftarrow \mathbf{x}^{temp}$, $r = r + 1$;
return $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{r+1}$.

We repeat the steps from Section V-B to Section V-F until there is a reduction in the ML cost. The complete procedure is provided in Algorithm 1. In the algorithm, $\mathbf{w}_{1:L-1,L}$ denotes the vector constituting the first $(L-1)$ elements of the Lth column of $\mathbf{W}$. Similarly $\mathbf{W}_{1:L,1:L}$ represents the matrix containing the common elements of the first L rows and the first L columns of $\mathbf{W}$, and $w_{L,L}$ represents the (L, L)th element of $\mathbf{W}$.

VI. Complexity of the Proposed Algorithm

We need to examine the complexity of computing the candidate sequence $\mathcal{T}$, the updates $\mathbf{p}_{(L,I_k)}^{(r)^*}$ (15) and $\Delta\phi_{(L,I_k)}^{(r)}$ (17).

The complexity of $\mathcal{T}$ depends on the computations required for determining $\mathbf{f}^{(r)}$ and $\mathbf{W}$, which are of the order of $O(N_r N_t)$ and $O(N_r N_t^2)$, respectively.

For the computation of $\mathbf{p}^{(r)^*}_{(L,I_k)}$ and $\Delta\phi^{(r)}_{(L,I_k)}$, we need to compute $\mathbf{W}_L^{-1}$ (for convenience here we omit the second subscript I_k). The symmetric structure of $\mathbf{W}_L$ allows us to compute $\mathbf{W}_{L+1}^{-1}$ as a function of $\mathbf{W}_L^{-1}$. If a symmetric matrix $\mathbf{W}_{L+1}$ can be expressed in terms of another symmetric matrix $\mathbf{W}_L$ as

$$\mathbf{W}_{L+1} = \begin{bmatrix} \mathbf{W}_L & \mathbf{w}_{1:L,L+1} \\ \mathbf{w}_{L+1,1:L} & w_{L+1,L+1} \end{bmatrix}, \tag{23}$$

and $\mathbf{W}_L^{-1}$ exists, then $\mathbf{W}_{L+1}^{-1}$ can be written in terms of $\mathbf{W}_L^{-1}$ as follows

$$\mathbf{W}_{L+1}^{-1} = \begin{bmatrix} \mathbf{W}_L^{-1} + \frac{1}{\rho}\mathbf{W}_L^{-1}\mathbf{b}\mathbf{b}^T\mathbf{W}_L^{-1} & -\frac{1}{\rho}\mathbf{W}_L^{-1}\mathbf{b} \\ -\frac{1}{\rho}\mathbf{b}^T\mathbf{W}_L^{-1} & \frac{1}{\rho} \end{bmatrix}, \tag{24}$$

where $\rho = w_{L+1,L+1} - \mathbf{b}^T\mathbf{W}_L^{-1}\mathbf{b}$ and $\mathbf{b} = \mathbf{w}_{1:L,L+1} = \mathbf{w}_{L+1,1:L}^T$. This has a computational complexity of the order of $O(L^2)$. Since $\mathbf{p}^{(r)^*}_{(L,I_k)}$ and $\Delta\phi^{(r)}_{(L,I_k)}$ need to be computed multiple times in each iteration, the overall computations required for finding $\mathbf{p}^{(r)^*}_{(L,I_k)}$ and $\Delta\phi^{(r)}_{(L,I_k)}$ $\forall L = 1, 2, \ldots, 2N_t$ is of the order of $O(N_t^3)$. Adding all the computations, the per iteration complexity of ULAS is of the order of $O(N_r N_t^2)$, where $N_r \geq N_t$.

Although the per iteration complexity of ULAS is less compared to $O(M^{1/2}N_r N_t^2)$ of 1-LAS and $O(M^{3/2}N_r N_t^4)$ of MLAS, the overall complexity cannot be determined only on the basis of per iteration complexity. This is because the overall complexity also depends on the number of iterations. We will continue this discussion in Section IX.

VII. Improving the Performance of ULAS Using Lattice Reduction

We know that the performance of linear detectors can be improved by conditioning the channel matrix $\mathbf{H}$ using lattice reduction (LR) techniques [37]. Since the expression for the update in (15) has a form similar to a linear detector, we expect an LR-aided extension of the ULAS algorithm to improve its error performance. In fact, we will see in Section IX that this can significantly improve the error performance, especially for large constellations and high values of SNR, with little increase in complexity.

A. Structure of the Proposed LR-Aided ULAS Algorithm

For a given basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, a lattice is defined as the linear combination of basis vectors as

$$\mathbb{L} = \left\{ \sum_{i=1}^{n} \psi_i \mathbf{b}_i | \psi_i \in \mathbb{Z} \right\}. \tag{25}$$

As the information symbols in (2) are drawn from an integer set Ω, we can see that $\mathbf{Hx} \in \mathbb{L}$, where the columns of $\mathbf{H}$ constitute the basis of $\mathbb{L}$.

An LR technique reduces the basis of $\mathbf{H}$ to find a more orthogonal basis $\widetilde{\mathbf{H}}$ while the lattice $\mathbb{L}$ remains the same for both the bases. Reducing the basis is equivalent to finding a unimodular matrix $\mathbf{T}$ such that $\widetilde{\mathbf{H}} = \mathbf{HT}$. For a unimodular matrix $\mathbf{T}$, all the entries of $\mathbf{T}$ and $\mathbf{T}^{-1}$ are Gaussian integers and the determinant of $\mathbf{T}$ is ± 1 [27].

Utilizing the new conditioned channel matrix $\widetilde{\mathbf{H}}$, (2) can be modeled as

$$\mathbf{y} = \widetilde{\mathbf{H}}\widetilde{\mathbf{x}} + \mathbf{n}, \tag{26}$$

where $\widetilde{\mathbf{x}} = \mathbf{T}^{-1}\mathbf{x}$ and the corresponding cost function (5) can be expressed as $\phi(\widetilde{\mathbf{x}}) = \|\mathbf{y} - \widetilde{\mathbf{H}}\widetilde{\mathbf{x}}\|^2$. Similar to Section IV, we consider the solution vector at the rth iteration as $\widetilde{\mathbf{x}}^{(r)}$ and the update vector as $\widetilde{\mathbf{p}}^{(r)}$. Thus, the solution vector for the $(r+1)$th iteration can be expressed as $\widetilde{\mathbf{x}}^{(r+1)} = \widetilde{\mathbf{x}}^{(r)} + \widetilde{\mathbf{p}}^{(r)}$ and the difference between the cost of two consecutive iterations is given by

$$\begin{aligned} \Delta\phi &= \phi(\widetilde{\mathbf{x}}^{(r+1)}) - \phi(\widetilde{\mathbf{x}}^{(r)}) \\ &= -2(\mathbf{y} - \widetilde{\mathbf{H}}\widetilde{\mathbf{x}}^{(r)})^T\widetilde{\mathbf{H}}\widetilde{\mathbf{p}}^{(r)} + \widetilde{\mathbf{p}}^{(r)^T}\widetilde{\mathbf{H}}^T\widetilde{\mathbf{H}}\widetilde{\mathbf{p}}^{(r)} \\ &= \widetilde{\mathbf{p}}^{(r)^T}\widetilde{\mathbf{W}}\widetilde{\mathbf{p}}^{(r)} - 2\widetilde{\mathbf{f}}^{(r)}\widetilde{\mathbf{p}}^{(r)}, \end{aligned} \tag{27}$$

where $\widetilde{\mathbf{W}} \triangleq \widetilde{\mathbf{H}}^T\widetilde{\mathbf{H}} = \mathbf{T}^T\mathbf{W}\mathbf{T}$ and $\widetilde{\mathbf{f}}^{(r)} \triangleq (\mathbf{y} - \widetilde{\mathbf{H}}\widetilde{\mathbf{x}}^{(r)})^T\widetilde{\mathbf{H}} = (\mathbf{y} - \mathbf{H}\mathbf{x}^{(r)})^T\widetilde{\mathbf{H}} = \mathbf{f}^{(r)}\mathbf{T}$. Proceeding on similar arguments as from (10) to (14), for a fixed L and I_k, the LR-aided RG update corresponding to (15) is given by

$$\widetilde{\mathbf{p}}^*_{(L,I_k)} = \widetilde{\mathbf{W}}^{-1}_{(L,I_k)}\widetilde{\mathbf{f}}^{(r)^T}_{(L,I_k)}. \tag{28}$$

However, we need to map this update to the original $\mathbf{H}$ so that it can be used in ULAS. To address this issue, we first introduce $\mathbf{1}_{N_t\times 1}$ to denote an $N_t \times 1$ vector with all entries being unity. Applying scaling and shifting on $\mathbf{x}$, as $\mathbf{x}_s = (\mathbf{x} - \mathbf{1}_{N_t\times 1})/2$, we map the elements of the constellation set Ω to a consecutive integer set, which are further mapped to the lattice-reduced domain as $\widetilde{\mathbf{x}}_s = \mathbf{T}^{-1}(\mathbf{x} - \mathbf{1}_{N_t\times 1})/2$, the elements of which are in the same consecutive integer set. The update corresponding to this can be expressed as

$$\begin{aligned} \widetilde{\mathbf{p}}_s^{(r)} &= \widetilde{\mathbf{x}}_s^{(r+1)} - \widetilde{\mathbf{x}}_s^{(r)} = \mathbf{T}^{-1}\left(\frac{\mathbf{x}^{(r+1)} - \mathbf{x}^{(r)}}{2}\right) \\ &= \frac{\mathbf{T}^{-1}\mathbf{p}^{(r)}}{2} \quad or \quad \frac{\widetilde{\mathbf{p}}^{(r)}}{2}. \end{aligned} \tag{29}$$

Combining (28) and (29) and applying the boundary conditions we get the following relationship

$$\mathbf{p}^*_{(L,I_k)} = 2\mathbf{T}_{(L,I_k)}\left\lceil \frac{\widetilde{\mathbf{W}}^{-1}_{(L,I_k)}\widetilde{\mathbf{f}}^{(r)^T}_{(L,I_k)}}{2} \right\rfloor, \tag{30}$$

where $\lceil\cdot\rfloor$ denotes the rounding operation to the nearest integer. We use this LR-aided RG update (30) to construct an LR-aided ULAS algorithm. In this algorithm except for the step described in Section V-E, all the other steps remain unchanged. In Section V-E we computed the RG update $\mathbf{p}^{(r)^*}_{(L,I_k)}$ and the rounding error $\boldsymbol{\delta}^{(r)^*}_{(L,I_k)}$ using $\mathbf{W}_{(L,I_k)}$ and $\mathbf{f}^{(r)}_{(L,I_k)}$ which are generated by sorting $\mathbf{W}$ and $\mathbf{f}^{(r)}$ respectively. Now, in the LR-aided version we need to first find the unimodular matrix $\mathbf{T}_{(L,I_k)}$ and then proceed similarly. In principle, for finding $\mathbf{T}_{(L,I_k)}$ we can use any of the available LR techniques on the first L columns of the $\mathbf{H}$ sorted on the basis of the candidate sequence $\mathcal{T}$.

Before we proceed, it may be noted that the algorithms which update one symbol per iteration, such as 1-LAS [14]

and RTS [16], [19] are not amenable to an LR-aided extension. This is because the expression for one symbol update will use only one column of $\mathbf{H}$ and conditioning of a single column of $\mathbf{H}$ is meaningless. Algorithms which update multiple symbols per iteration, such as MLAS [15] are limited to two or three symbol updates only. This means an LR-aided extension will require conditioning of two or three columns of $\mathbf{H}$, which is again meaningless for large size matrices. Hypothetically speaking, even if one could somehow address the issue of conditioning of one-three columns of $\mathbf{H}$, an LR-aided extension will still not be possible until the counterpart of (30) is found. This again seems difficult.

Coming back to the LR-aided ULAS algorithm, we need to compute the best LR-aided RG update (30) in an L symbol neighborhood for $L = 1$ to $2N_t$. However, a straightforward application of the LR technique will lead to a significant increase in complexity. Therefore, there is a need to modify the existing LR techniques so that the available information at Lth iteration can be used to reduce the computations at $(L+1)$th iteration. We address this issue in the next section.

VIII. Adapting Element Based Lattice Reduction for ELR-Aided ULAS

In the literature there exist several lattice reduction techniques such as Gaussian reduction [39], Lenstra-Lenstra-Lovász (LLL) reduction [40], Seysen reduction [41], Minkowski [42], and Hermite-Korkine-Zolotareff reduction [43], [44]. However, when the problem size is large, these LR techniques either have high computational complexity [41], [45] or do not perform well [46], [47]. This has been taken care of in the element based lattice reduction (ELR) technique [27] which is an extension of [47].

The prime motive of these LR techniques is to transform the channel matrix $\mathbf{H}$ to $\widetilde{\mathbf{H}}$ such that the orthogonal deficiency (OD) [29] is reduced. Here OD is a number between 0 to 1 and is defined as

$$\mathrm{OD}(\mathbf{H}) = 1 - \frac{\det(\mathbf{H}^H\mathbf{H})}{\prod_{i=1}^{2N_t} \|\mathbf{h}_i\|^2}. \tag{31}$$

Thus, for an orthogonal matrix OD is 0 while it is 1 for a singular matrix.

The ELR technique attempts to improve the diagonal structure of $\widetilde{\mathbf{H}}^H\widetilde{\mathbf{H}}$ compared to $\mathbf{H}^H\mathbf{H}$. Two versions of ELR have been proposed in the literature viz. primal ELR and dual ELR. A primal ELR operates on $\mathbf{H}^H\mathbf{H}$ while the dual operates on $(\mathbf{H}^H\mathbf{H})^{-1}$. First we discuss primal ELR based ULAS.

A. Modified Primal ELR-Aided ULAS (MPELR-ULAS)

Without loss of generality, we select the first L columns of $\mathbf{H}$ and denote it by $\mathbf{H}_L$. If we know $\widetilde{\mathbf{W}}_L$, $\widetilde{\mathbf{H}}_L$, and $\mathbf{T}_L$, we can determine the initial values of $\widetilde{\mathbf{W}}_{L+1}$ and $\mathbf{T}_{L+1}$ for the conditioning of $\mathbf{H}_{L+1}$, as follows

$$\widetilde{\mathbf{W}}_{L+1} = \begin{bmatrix} \widetilde{\mathbf{W}}_L & \widetilde{\mathbf{H}}_L^T\mathbf{h}_{L+1} \\ \mathbf{h}_{L+1}^T\widetilde{\mathbf{H}}_L & \|\mathbf{h}_{L+1}\|^2 \end{bmatrix}, \quad \mathbf{T}_{L+1} = \begin{bmatrix} \mathbf{T}_L & 0 \\ 0 & 1 \end{bmatrix}. \tag{32}$$

Now using the ELR method described in [27] the following row and column operations can be performed

$$(\mathbf{T}_{L+1})_k = (\mathbf{T}_{L+1})_k + \lambda_{i,k}(\mathbf{T}_{L+1})_i, \tag{33}$$

Algorithm 2 Modified Dual Element Lattice Reduction

Input : $\mathbf{H}$
Output: $\widetilde{\mathbf{H}}_L$, $\mathbf{T}_L$, $\widetilde{\mathbf{W}}_L^{-1}$ $\forall L = 1, 2, \ldots, 2N_t$

Initialization $\widetilde{\mathbf{H}}_1 = \mathbf{h}_1$, $\mathbf{T}_1 = 1$, $\widetilde{\mathbf{W}}_1^{-1} = \left(\|\mathbf{h}_1\|^2\right)^{-1}$, $\mathbf{H}_{sel} = \mathbf{h}_1$;
for $L \leftarrow 1$ **to** $2N_t - 1$ **do**
  $\mathbf{b} = \widetilde{\mathbf{H}}_L^T\mathbf{h}_{L+1}$;
  $\rho = \|\mathbf{h}_{L+1}\|^2 - \mathbf{b}^T\widetilde{\mathbf{W}}_L^{-1}\mathbf{b}$;
  $\widetilde{\mathbf{W}}_{L+1}^{-1} = \begin{bmatrix} \widetilde{\mathbf{W}}_L^{-1} + \frac{1}{\rho}\widetilde{\mathbf{W}}_L^{-1}\mathbf{b}\,\mathbf{b}^T\widetilde{\mathbf{W}}_L^{-1} & -\frac{1}{\rho}\widetilde{\mathbf{W}}_L^{-1}\mathbf{b} \\ -\frac{1}{\rho}\mathbf{b}^T\widetilde{\mathbf{W}}_L^{-1} & \frac{1}{\rho} \end{bmatrix}$,
  $\mathbf{T}_{L+1} = \begin{bmatrix} \mathbf{T}_L & 0 \\ 0 & 1 \end{bmatrix}$;
  $\mathbf{H}_{sel} = [\mathbf{H}_{sel} \;\; \mathbf{h}_{L+1}]$;
  $u = 1$;
  while $(u \le \kappa)$ **do**
    $k \leftarrow$ select the index of uth largest diagonal element of $\widetilde{\mathbf{W}}_{L+1}$;
    $\lambda_{i,k} = -\left\lfloor \frac{\widetilde{w}_{i,k}}{\widetilde{w}_{i,i}} \right\rceil$ $\forall i \neq k$;
    if $\lambda_{i,k} = 0 \;\; \forall i \neq k$ **then**
      $u \leftarrow u + 1$;
    else
      $\Delta_{i,k} = -|\lambda_{i,k}|^2\widetilde{w}_{i,i}^2 - \lambda_{i,k}^*\widetilde{w}_{i,k} - \lambda_{i,k}\widetilde{w}_{k,i}$;
      Choose $i = \underset{\tilde{i}\neq k}{\operatorname{argmax}}\, \Delta_{\tilde{i},k}$;
      $(\widetilde{\mathbf{W}}_{L+1}^{-1})_k \leftarrow (\widetilde{\mathbf{W}}_{L+1}^{-1})_k + \lambda_{i,k}(\widetilde{\mathbf{W}}_{L+1}^{-1})_i$;
      $(\widetilde{\mathbf{W}}_{L+1}^{-1})^k \leftarrow (\widetilde{\mathbf{W}}_{L+1}^{-1})^k + \lambda_{i,k}^*(\widetilde{\mathbf{W}}_{L+1}^{-1})^i$;
      $(\mathbf{T}_{L+1})_i \leftarrow (\mathbf{T}_{L+1})_i - \lambda_{i,k}^*(\mathbf{T}_{L+1})_k$;
  $\widetilde{\mathbf{H}}_{L+1} = \mathbf{H}_{sel}\mathbf{T}_{L+1}$;
return $\widetilde{\mathbf{H}}_L$, $\mathbf{T}_L$, and $\widetilde{\mathbf{W}}_L^{-1}$ $\forall L = 1, 2, \ldots, 2N_t$.

$$(\widetilde{\mathbf{W}}_{L+1})_k = (\widetilde{\mathbf{W}}_{L+1})_k + \lambda_{i,k}(\widetilde{\mathbf{W}}_{L+1})_i, \tag{34}$$

$$(\widetilde{\mathbf{W}}_{L+1})^k = (\widetilde{\mathbf{W}}_{L+1})^k + \lambda_{i,k}^*(\widetilde{\mathbf{W}}_{L+1})^i, \tag{35}$$

where $(\cdot)_k$ and $(\cdot)^k$ represent kth column and kth row respectively, and $\lambda_{i,k}$ is a constant given by

$$\lambda_{i,k} = -\left\lfloor \frac{\widetilde{w}_{i,k}}{\widetilde{w}_{i,i}} \right\rceil \quad \forall i \neq k, \tag{36}$$

where k is the index of the uth largest diagonal element $(u = 1, 2, \ldots, \kappa)$ and i is the index $(\neq k)$ for which the following expression is maximized

$$-|\lambda_{i,k}|^2\widetilde{w}_{i,i}^2 - \lambda_{i,k}^*\widetilde{w}_{i,k} - \lambda_{i,k}\widetilde{w}_{k,i}. \tag{37}$$

These row and column operations can be performed in either of two ways: until the largest diagonal elements become irreducible (this corresponds to $\kappa = 1$) or all the diagonal elements become irreducible (this corresponds to $\kappa = 2N_t$). A diagonal element $\widetilde{w}_{k,k}$ is reducible if and only if there exists $i \neq k$, such that $\lambda_{i,k} \neq 0$. We call this algorithm modified primal ELR (MPELR). The version of ULAS which incorporates MPELR is referred to as MPELR-ULAS.

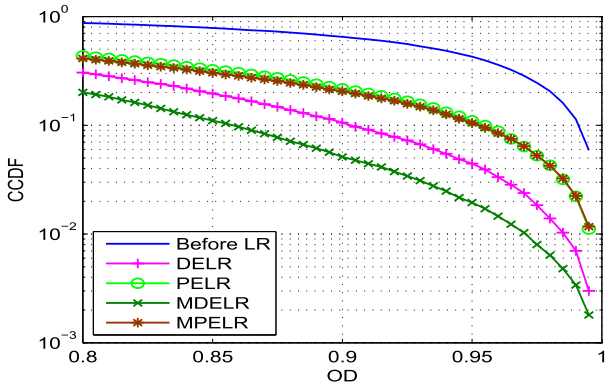

Fig. 1. CCDF plot of OD for different ELR techniques for 4×4 channel matrix $\bar{\mathbf{H}}$.

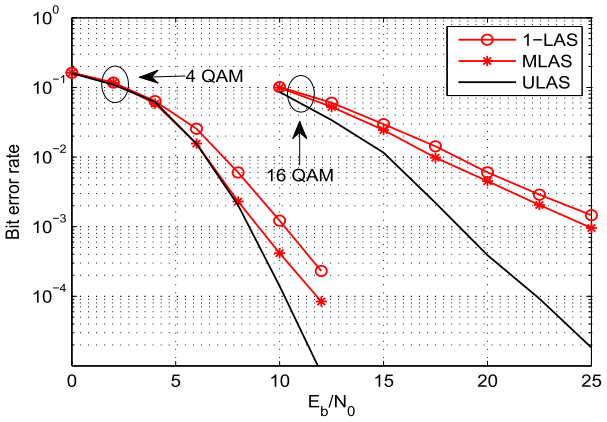

Fig. 2. BER of a 32×32 MIMO system for 4-QAM and 16-QAM.

B. Modified Dual ELR-Aided ULAS (MDELR-ULAS)

Similar to the primal problem, a dual problem which is based on the inverse of $\mathbf{W}$ i.e. $(\mathbf{H}^H\mathbf{H})^{-1}$ is also addressed in [27]. Since LR-aided ULAS requires matrix inversion [see (30)], application of dual ELR to ULAS is expected to be more beneficial than the application of primal ELR to ULAS. The pseudo code of the modified dual ELR algorithm is shown as Algorithm 2 and referred to as MDELR. It is similar to ELR except for the row and column operations. We call the MDELR aided version of ULAS as MDELR-ULAS.

We expect the OD values resulting from modified ELR techniques to be better than the values obtained from the respective ELR techniques. This is because the modified ELR obtains the conditioned matrix of order $L+1$ from the conditioned matrix of order L. Let us examine the OD values before and after the application of modified ELR techniques and compare them with the respective ELR techniques. The complementary CDF (CCDF) of OD values is shown in Fig. 1. We find that MPELR has a slightly better OD compared to the original primal ELR while MDELR has a significantly better OD compared to the original dual ELR. From the complexity viewpoint also, it may be pointed out that MDELR directly computes $\widetilde{\mathbf{W}}_L^{-1}$ while in the case of MPELR we first obtain $\widetilde{\mathbf{W}}_L$ and subsequently compute $\widetilde{\mathbf{W}}_L^{-1}$. This means that MDELR is better than MPELR in terms of OD as well as complexity. In other words, MDELR lends itself to a better adaptation of ELR technique compared to MPELR.

IX. Simulation Results

In this section, we study the performance of the proposed ULAS algorithm with and without LR techniques through extensive simulations and compare the bit error rate (BER) performance and the required per bit computations with the existing algorithms. We begin with the comparison of ULAS with different variants of LAS taking MMSE solution as the initial vector.

A. Comparison of ULAS With Variants of LAS Algorithms

We compare the proposed ULAS algorithm with 1-LAS and MLAS for 32×32 and 64×64 MIMO systems for 4-QAM (quadrature amplitude modulation) and 16-QAM modulations. The bit error performance is shown in Fig. 2 and Fig. 4, while

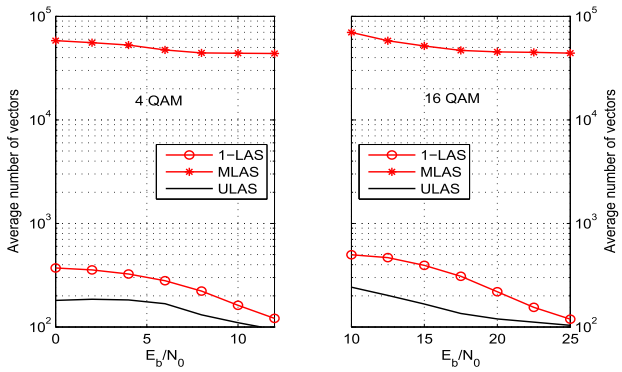

Fig. 3. Average number of vectors searched in a 32×32 MIMO system with 4-QAM and 16-QAM.

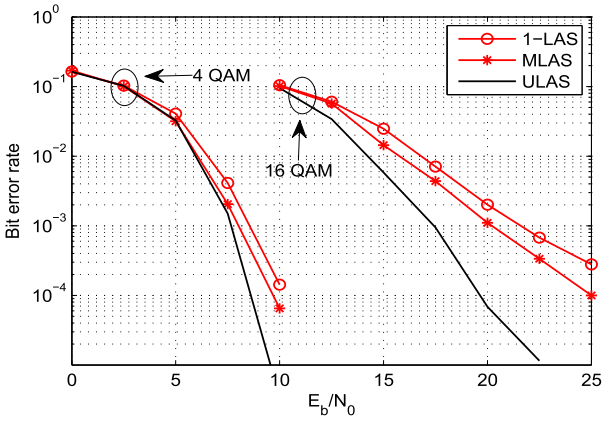

Fig. 4. BER of a 64×64 MIMO system for 4-QAM and 16-QAM.

the number of vectors searched (as a measure of complexity) is shown in Fig. 3 and Fig. 5. From the figures, it can be observed that the proposed ULAS not only improves the error performance but also searches a lower number of vectors compared to both the variants of LAS.

For example, compared to 1-LAS, at 20 dB ULAS requires 8 dB less E_b/N_0 to achieve a BER of 10^{-3} (see Fig. 2) and searches 40% less vectors for a 32×32 MIMO system using 16-QAM modulation. Compared to MLAS, for the same system and BER, ULAS requires 7 dB less E_b/N_0 and more than 99% reduction in the number of vectors searched. However,

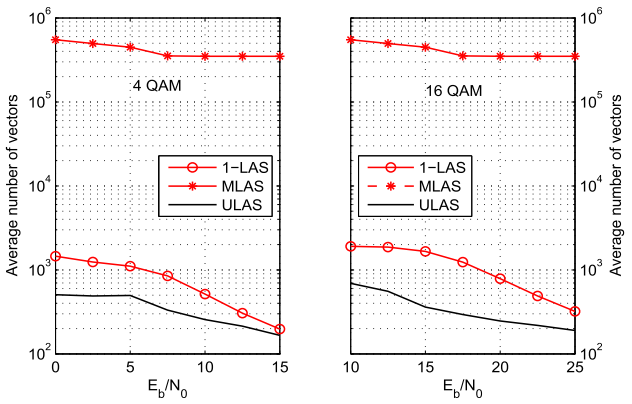

Fig. 5. Average number of vectors searched in a 64 × 64 MIMO system with 4-QAM and 16-QAM.

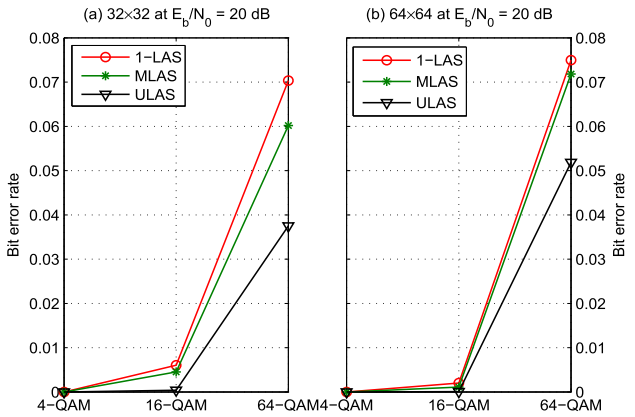

Fig. 6. BER of 32 × 32 and 64 × 64 MIMO systems at $E_b/N_0 = 20$ dB.

TABLE I

COMPLEXITY OF 32 × 32 AND 64 × 64 MIMO SYSTEMS FOR 4-QAM AND 16-QAM

Detection Algorithms	Total number of arithmetic operations ($\times 10^4$) per bit (number of iterations) required for 32 × 32 MIMO system			
	4-QAM		16-QAM	
	$E_b/N_0 = 5$ dB	$E_b/N_0 = 10$ dB	$E_b/N_0 = 15$ dB	$E_b/N_0 = 20$ dB
MMSE	0.854	0.854	0.427	0.427
1-LAS	6.172 (4.8)	3.213 (2.5)	3.903 (6.3)	2.169 (3.4)
MLAS	619.1 (7.5)	574.5 (4.6)	338.2 (9.8)	295.6 (5.7)
ULAS	4.053 (2.6)	2.655 (1.7)	2.014 (2.4)	1.445 (1.8)
64 × 64 MIMO system				
MMSE	3.347	3.347	1.673	1.673
1-LAS	43.201 (8.6)	20.104 (3.9)	32.456 (13.2)	15.368 (6.3)
MLAS	11578.1 (13.1)	9040.97 (6.3)	7414.49 (16.2)	5798.71 (8.1)
ULAS	23.612 (3.6)	12.194 (1.9)	8.602 (2.8)	5.864 (1.9)

in general, the number of vectors searched is not enough to convey the overall complexity. Hence we compute the total number of arithmetic operations i.e. a product of per iteration complexity and the number of iterations. We have already discussed per iteration complexity in Section VI. In Table I, we provide the total number of arithmetic operations, along with the average number of iterations in small brackets. One can observe that ULAS requires less average number of iterations and has lower per iteration complexity, compared to both 1-LAS and MLAS algorithms. This results in lower number of arithmetic operations, overall. Thus, for the same system, i.e. a 32×32 MIMO system with 16-QAM, at 20 dB there is a 34% reduction in the number of arithmetic operations as compared to 1-LAS while there is more than 99% reduction compared to MLAS. The trend for BER gain and savings in complexity (in terms of the number of vectors searched and the number of arithmetic operations) have been found to be similar for other systems. Thus we can conclude that the proposed ULAS is significantly better in terms of error performance as well as complexity, compared to 1-LAS and MLAS.

Further, we examine the relative performance of ULAS, 1-LAS, and MLAS for increasing constellation size. The results have been shown in Fig. 6. From the figure, one can observe that for 32 × 32 and 64 × 64 MIMO systems at 20 dB the proposed ULAS provides significant improvement as compared to 1-LAS as well as MLAS for 4-QAM, 16-QAM and 64-QAM constellations. In fact, the BER gain increases with increasing constellation size.

We have also examined the error performance as a function of the number of antennas. The simulation results are shown

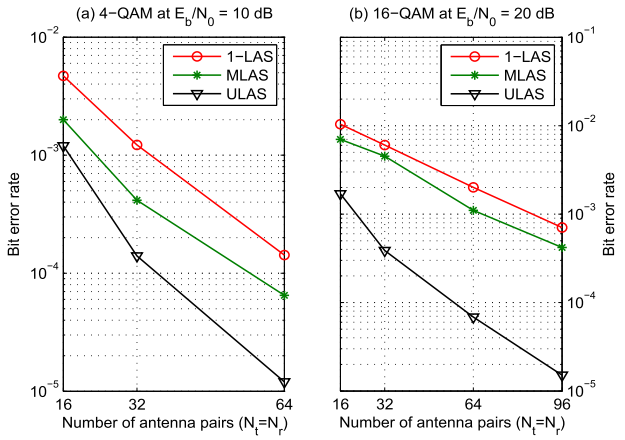

Fig. 7. BER with increasing number of antennas for 4-QAM at $E_b/N_0 = 10$ dB and for 16-QAM at $E_b/N_0 = 20$ dB.

in Fig. 7 for 4-QAM at 10 dB and 16-QAM at 20 dB. From the results one can see that to achieve the same BER, ULAS requires less number of antennas compared to 1-LAS and MLAS. For example, for a 16-QAM system at 20 dB, ULAS can achieve a BER of 4×10^{-4} using only 32 antenna pairs while MLAS will require 96.

B. Understanding the Superior Performance of ULAS

The reason for the better performance of ULAS is that the number of symbols differing between two consecutive iterations is adaptive, unlike 1-LAS and MLAS. This facilitates two things, firstly an adaptive change per iteration provides an opportunity for faster convergence, thereby lowering complexity. Secondly, because of the faster convergence, the early termination problem is reduced, resulting in an improved error performance. To corroborate both these assertions, we examine the average value of L i.e. the average number of differing symbols between two consecutive iterations. From Table II, one can observe that the average value of L is higher for ULAS as compared to 1-LAS and MLAS. We compare these numbers with the average number of symbols differing between the initial MMSE solution vectors which are erroneous and the original vectors. One can observe that because of a higher average value of L, ULAS requires a lower number of iterations to converge to the solution, and thus saves complexity. A smaller average value of L (as in the case of 1-LAS and MLAS)

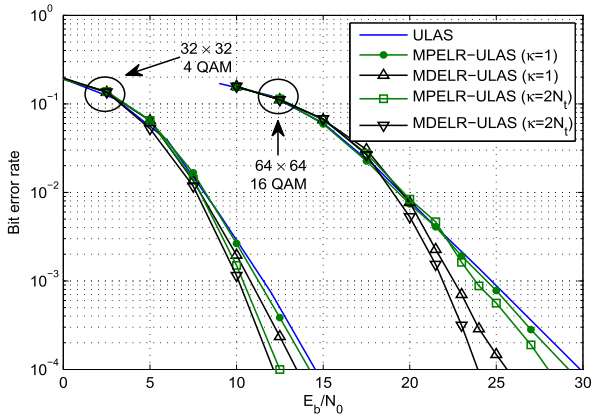

Fig. 8. BER comparison of ULAS with its LR-aided extensions for a 32 × 32 MIMO system with 4-QAM and a 64 × 64 MIMO system with 16-QAM.

TABLE II

AVERAGE VALUE OF L PER ITERATION AND THE AVERAGE NUMBER OF SYMBOLS DIFFERING FROM ERRONEOUS INITIAL MMSE SOLUTIONS IN A 64 × 64 MIMO SYSTEM FOR 4-QAM AND 16-QAM

Detection Algorithms	Average value of L per iteration			
	64 × 64, 4-QAM		64 × 64, 16-QAM	
	$E_b/N_0 = 5$ dB	$E_b/N_0 = 10$ dB	$E_b/N_0 = 15$ dB	$E_b/N_0 = 20$ dB
1-LAS	1	1	1	1
MLAS	1.3174	1.5184	1.2364	1.4234
ULAS	2.3892	1.8866	5.0569	2.5379
Number of symbols differing from erroneous initial MMSE solutions				
Transmitted vector $\mathbf{x}$	9.1840	3.3696	13.2678	5.7904

requires a larger number of iterations to converge. Further, as stated above, this sometimes leads to early termination causing a loss in error performance.

C. Comparison With ELR-Aided Variants of ULAS

Having established the superiority of ULAS over all other LAS algorithms, in this subsection we examine the performance of the proposed MDELR-ULAS and MPELR-ULAS taking ZF solution as the initial vector (refer Section VIII). We consider two extreme cases $\kappa = 1$ and $\kappa = 2N_t$, thus leading to four variants overall. BER results are shown in Fig. 8 while the total number of arithmetic operations per bit is provided in Table III. From these results, we observe that the ELR-aided algorithms have an improved error performance at the expense of additional computations. The additional computations required by these four variants are similar, as can be seen from Table III. However, the MDELR-ULAS with $\kappa = 2N_t$ is found to be best in terms of improvement in error performance. For example, to achieve a BER of 10^{-4}, MDELR-ULAS with $\kappa = 2N_t$ has a 6 dB gain compared to ULAS for a 64 × 64 MIMO system with 16-QAM modulation while there is more than 2 dB gain for a 32 × 32 MIMO system with 4-QAM modulation (Fig. 8). Keeping in view its significant gain in terms of error performance, the MDELR-ULAS with $\kappa = 2N_t$ is the best among these four variants, and we use it for comparison with other existing detectors, in the next subsection.

D. Comparison With Other Existing Detectors

Now we compare the performance of the proposed ULAS and MDELR-ULAS ($\kappa = 2N_t$) with the existing popular detectors for large MIMO systems [16]–[19], [24], [26]–[28], [30] beyond the LAS family of algorithms. These algorithms can be grouped into three different categories, i.e. neighborhood search algorithms [16]–[19], algorithms based on LR techniques [27], and sparse error recovery based algorithms [24], [26]. We have chosen three recent algorithms, one from each category i.e. RTS, ELR-MMSE [27], and sparsity boosted iterative linear detector (SBIL) [24]. These are the best known algorithms in their respective categories. However, we have skipped comparison with LTS though it is true that LTS can provide better error performance compared to RTS, due to its very high complexity. The simulation parameters for RTS and SBIL are taken to be same as in [19] and [24] respectively. We have shown the comparison for 32 × 32 and 64 × 64 MIMO systems with 64-QAM modulation. The bit error performance is shown in Fig. 9 and Fig. 11, respectively while the total number of arithmetic operations required is shown in Fig. 10 and Fig. 12, respectively. For completeness, we have also compared with 1-LAS, MLAS, and MMSE receivers.

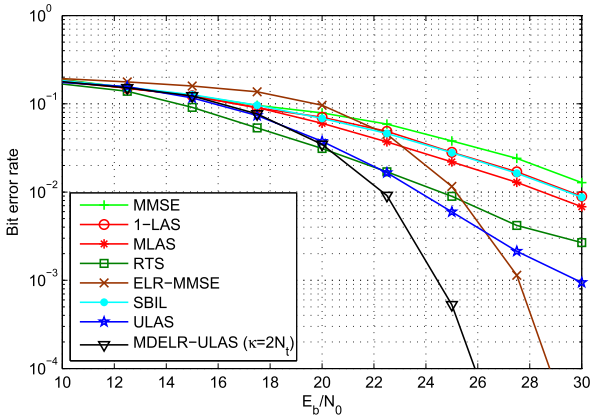

Fig. 9. BER of a 32 × 32 MIMO system for 64-QAM.

TABLE III

COMPLEXITY COMPARISON OF ULAS WITH ITS LR-AIDED EXTENSIONS FOR A 32 × 32 MIMO SYSTEM WITH 4-QAM AND A 64 × 64 MIMO SYSTEM WITH 16-QAM

Detection Algorithms	Total number of arithmetic operations per bit ($\times 10^4$) required			
	32 × 32, 4-QAM		64 × 64, 16-QAM	
	$E_b/N_0 = 10$ dB	$E_b/N_0 = 15$ dB	$E_b/N_0 = 15$ dB	$E_b/N_0 = 20$ dB
ULAS	3.774	2.848	17.765	9.762
MPELR-ULAS ($\kappa = 1$)	48.37	36.41	424.9	238.1
MDELR-ULAS ($\kappa = 1$)	47.76	36.38	426.9	218.1
MPELR-ULAS ($\kappa = 2N_t$)	49.12	36.68	421.4	250.7
MDELR-ULAS ($\kappa = 2N_t$)	48.09	38.05	448.2	223.9

From Fig. 9 we can see that for a 32×32 MIMO system with 64-QAM modulation, the MDELR-ULAS with $\kappa = 2N_t$ has a uniformly better error performance compared to all the others, except RTS. RTS has a slightly better error performance than MDELR-ULAS up to a BER of $10^{-1.5}$ and up to a BER of $10^{-1.7}$ compared to ULAS. After these BER values, MDELR-ULAS and ULAS have a better BER than RTS. However, the total number of arithmetic operations required by RTS is around 100 times higher as compared to MDELR-ULAS while it is approximately 1000 times compared to ULAS (as shown in Fig. 10). In addition to this, one can notice that the rate of improvement in the BER with E_b/N_0 is much better for MDELR-ULAS as compared to RTS. Thus, from Fig. 9 we can observe that from $E_b/N_0 = 20$ dB to 25 dB, the BER of MDELR-ULAS

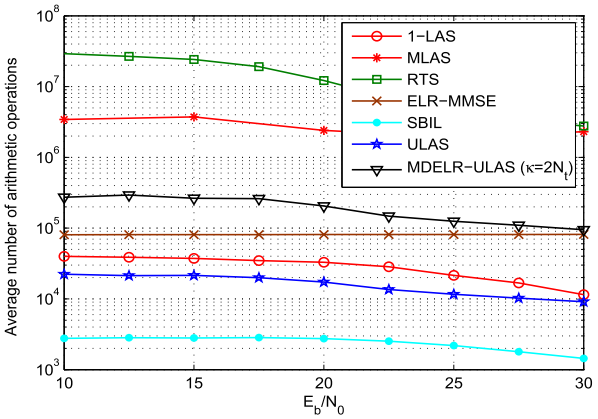

Fig. 10. Average number of arithmetic operations required in a 32×32 MIMO system with 64-QAM.

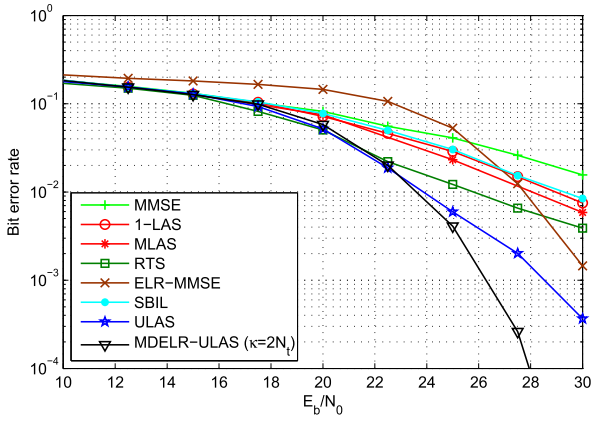

Fig. 11. BER of a 64×64 MIMO system for 64-QAM.

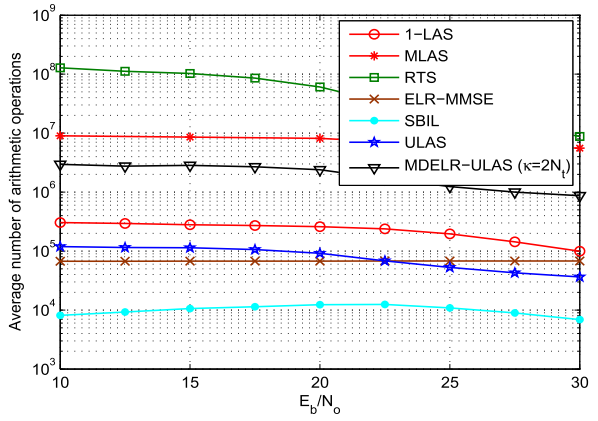

Fig. 12. Average number of arithmetic operations required in a 64×64 MIMO system with 64-QAM.

improves from 4×10^{-2} to 5×10^{-4} while in the case of RTS it improves from 3×10^{-2} to only 9×10^{-3}.

In the case of a 64×64 MIMO system with 64-QAM modulation, ULAS and MDELR-ULAS with $\kappa = 2N_t$, both have better error performance compared to all the others throughout the range of E_b/N_0 (shown in Fig. 11). From Fig. 11 and Fig. 12, we can see that except for ELR-MMSE and SBIL, all the others have not only poor error performance but also have a higher number of arithmetic operations. Only in the case of ELR-MMSE and SBIL, we have a lower number of arithmetic operations, but that comes at the cost of an inferior error performance (see Fig. 11).

As pointed out earlier in Section VII an LR-aided extension is not of much interest in the case of 1-LAS, MLAS, and RTS. However, an LR-aided SBIL may be worth exploring.

E. Discussion

The fundamental innovation in ULAS is to search adaptively instead of searching in a fixed neighborhood. Since the overall structure of ULAS is similar to LAS algorithms, it can easily incorporate the additional strategies used in neighborhood search algorithms to improve their error performance. For example, to avoid cycles or early terminations, strategies like the ones used in RTS can be used which can further be extended to layered schemes [19] too. Similar to [17], a search can also be initialized with multiple vectors.

X. Conclusion

In the framework of neighborhood search algorithms, we have proposed a metric to determine the likelihood of the erroneous locations and derived an expression for finding a reasonably good update at the selected erroneous locations. Unlike existing algorithms, the framework provides for an adaptive number of erroneous locations to be updated. The update which minimizes the cost function is selected at each iteration. The process continues until there is a reduction in the ML cost. Simulation results show that the proposed ULAS algorithm has much improved error performance as well as lesser computational complexity compared to the existing algorithms. Also, the rate of improvement with increasing number of antenna pairs is faster as compared to 1-LAS and MLAS algorithms. Lastly, we have derived two variants of ELR-aided ULAS. Simulation results show that the two variants provide an improved error performance at the expense of some additional computations.

References

[1] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[2] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA Implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[3] J. Mietzner, R. Schober, L. Lampe, W. Gerstacker, and P. Hoeher, "Multiple-antenna techniques for wireless communications—A comprehensive literature survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 87–105, 2nd Quart., 2009.

[4] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[5] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[6] E. Larsson, "MIMO detection methods: How they work [lecture notes]," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 91–95, May 2009.

[7] S. W. Kim and K. P. Kim, "Log-likelihood-ratio-based detection ordering in V-BLAST," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 302–307, Feb. 2006.

[8] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[9] S. R. Lee, S. H. Park, S. W. Kim, and I. Lee, "Enhanced detection with new ordering schemes for V-BLAST systems," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1648–1651, Jun. 2009.

[10] B. S. Thian and A. Goldsmith, "A reduced-complexity MIMO receiver via channel ordering," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Nov./Dec. 2003, pp. 1–6.

[11] C.-Y. Hung, R. Chang, and W.-H. Chung, "A hybrid MMSE and K-best detection scheme for MIMO systems," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2012, pp. 1–5.

[12] T. Cui, S. Han, and C. Tellambura, "Probability-distribution-based node pruning for sphere decoding," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1586–1596, May 2013.

[13] T.-H. Kim, "Low-complexity sorted QR decomposition for MIMO systems based on pairwise column symmetrization," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1388–1396, Mar. 2014.

[14] K. V. Vardhan, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 473–485, Apr. 2008.

[15] S. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2008, pp. 2012–2016.

[16] B. Rajan, S. Mohammed, A. Chockalingam, and N. Srinidhi, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2009, pp. 1993–1997.

[17] P. Li and R. Murch, "Multiple output selection-LAS algorithm in large MIMO systems," *IEEE Commun. Lett.*, vol. 14, no. 5, pp. 399–401, May 2010.

[18] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan, "Random-restart reactive tabu search algorithm for detection in large-MIMO systems," *IEEE Commun. Lett.*, vol. 14, no. 12, pp. 1107–1109, Dec. 2010.

[19] N. Srinidhi, T. Datta, A. Chockalingam, and B. Rajan, "Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 2955–2963, Nov. 2011.

[20] A. A. Pereira and R. Sampaio-Neto, "A random-list based LAS algorithm for near-optimal detection in large-scale uplink multiuser MIMO systems," in *Proc. 19th Int. ITG Workshop Smart Antennas (WSA)*, Mar. 2015, pp. 1–5.

[21] P. Som, T. Datta, A. Chockalingam, and B. Rajan, "Improved large-MIMO detection based on damped belief propagation," in *Proc. IEEE Workshop Inf. Theory (ITW)*, Jan. 2010, pp. 1–5.

[22] P. Som, T. Datta, N. Srinidhi, A. Chockalingam, and B. Rajan, "Low-complexity detection in large-dimension MIMO-ISI channels using graphical models," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 8, pp. 1497–1511, Dec. 2011.

[23] T. Narasimhan and A. Chockalingam, "EXIT chart based design of irregular LDPC codes for large-MIMO systems," *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 115–118, Jan. 2013.

[24] X. Peng, W. Wu, J. Sun, and Y. Liu, "Sparsity-boosted detection for large MIMO systems," *IEEE Commun. Lett.*, vol. 19, no. 2, pp. 191–194, Feb. 2015.

[25] Y. Fadlallah, A. Aïssa-El-Bey, K. Amis, D. Pastor, and R. Pyndiah, "New iterative detector of MIMO transmission using sparse decomposition," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3458–3464, Aug. 2015.

[26] J. W. Choi and B. Shim, "New approach for massive MIMO detection using sparse error recovery," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 3754–3759.

[27] Q. Zhou and X. Ma, "Element-based lattice reduction algorithms for large MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 274–286, Feb. 2013.

[28] K. Ali, W. Abediseid, and M.-S. Alouini, "Sequential decoders for large MIMO systems," in *Proc. 12th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014, pp. 709–716.

[29] X. Ma and W. Zhang, "Performance analysis for MIMO systems with lattice-reduction aided linear equalization," *IEEE Trans. Commun.*, vol. 56, no. 2, pp. 309–318, Feb. 2008.

[30] P. Svac, F. Meyer, E. Riegler, and F. Hlawatsch, "Soft-heuristic detectors for large MIMO systems," *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4573–4586, Sep. 2013.

[31] L. Dai, X. Gao, X. Su, S. Han, C.-L. I, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss-Seidel method for uplink multiuser large-scale MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4839–4845, Oct. 2015.

[32] A. Elghariani and M. Zoltowski, "Low complexity detection algorithms in large-scale MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1689–1702, Mar. 2016.

[33] A. K. Sah and A. K. Chaturvedi, "Improving the reliability of receivers for 5G networks," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2016, pp. 1–2.

[34] K. S. Gyamfi, J. Baek, and K. Lee, "Lattice reduction aided tabu search with channel-dependent stopping criterion for MIMO systems," *IET Electron. Lett.*, vol. 51, no. 24, pp. 2062–2064, Nov. 2015.

[35] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.

[36] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[37] D. Wubben, D. Seethaler, J. Jalden, and G. Matz, "Lattice reduction," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 70–91, May 2011.

[38] A. K. Sah and A. K. Chaturvedi, "Reduced neighborhood search algorithms for low complexity detection in MIMO systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[39] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, vol. 1. Nov. 2002, pp. 424–428.

[40] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, Dec. 1982.

[41] M. Seysen, "Simultaneous reduction of a lattice basis and its reciprocal basis," *Combinatorica*, vol. 13, no. 3, pp. 363–376, 1993.

[42] C. Hermite, "Extraits de lettres de M. Ch. Hermite á M. Jacobi sur diffrents objects de la th èorie des nombres," *J. Reine Angewandte Math.*, vol. 40, nos. 3–4, pp. 261–315, 1850.

[43] H. Minkowski, "Diskontinuitätsbereich für arithmetische Äquivalenz," *J. Reine Angewandte Math.*, vol. 129, nos. 3–4, pp. 220–274, 1905.

[44] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[45] W. Zhang, X. Ma, and A. Swami, "Designing low-complexity detectors based on Seysen's algorithm," *IEEE Trans. Wireless Commun.*, vol. 9, no. 10, pp. 3301–3311, Oct. 2010.

[46] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2795–2808, Jun. 2011.

[47] Q. Zhou and X. Ma, "Designing low-complexity detectors for generalized SC-FDMA systems," in *Proc. 45th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2011, pp. 1–6.

Abhay Kumar Sah (S'15) received the B.Tech. degree in electronics and communications engineering from the Jaypee University of Information Technology, Waknaghat, India, in 2010. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, IIT Kanpur, Kanpur, India. His current research interests are in wireless communication and multi-antenna systems. He was a recipient of the TCS Research Scholarship and the Vice-Chancellors Gold Medal for outstanding performance in his bachelor's degree.

A. K. Chaturvedi (S'91–M'96–SM'03) received the B.Tech., M.Tech., and Ph.D. degrees in electrical engineering from IIT Kanpur, Kanpur, in 1986, 1988, and 1995, respectively. He served the Department of Electronics Engineering, IIT Varanasi, Varanasi, from 1994 to 1996. He joined the faculty of the Department of Electronics and Computer Engineering, IIT Roorkee, Roorkee. In 1999, he joined IIT Kanpur, where he also held the positions of Head of the Department of Electrical Engineering, the Dean of Research and Development, and the Deputy Director. He was the Coordinator of the BSNL-IITK Telecom Centre of Excellence which has done a large number of projects for the Indian telecom sector. He is currently the Director of IIT Roorkee. His research interests are in communication theory and wireless communications. He was a recipient of the Distinguished Teacher Award of IIT Kanpur and a Tan Chin Tuan Fellowship of Nanyang Technical University, Singapore. He is a member of the Nomination and Election Committee of the IEEE Communication Society. He has also served on the Teaching Awards Committee of the IEEE. He is a Founding Member of the Telecom Standards Development Society of India and the current Chair of the Joint Telematics Group, which organizes the annual National Conference on Communications.