Beyond Fixed Neighborhood Search in the Likelihood Ascent Algorithm for MIMO Systems

Abhay Kumar Sah[†] and A. K. Chaturvedi[‡]
Department of Electrical Engineering
Indian Institute of Technology Kanpur, Kanpur, India 208016
Email: {abhaysah[†], akc[‡]}@iitk.ac.in

***Abstract*—Neighborhood search algorithms have been proposed for detection in large/massive multiple-input multiple-output (MIMO) systems. They iteratively search for the best vector in a fixed neighborhood. However, the ML solution may not lie in the searched space or the search may take a large number of intermediate vectors to converge. Instead of searching in a fixed neighborhood, a better way will be to look for an update which is not restricted to be in a fixed neighborhood. Motivated by this, we formulate an optimization problem to maximize the reduction in ML cost and use it to derive an expression for updating the solution. We use a metric based on the channel matrix and the error vector to determine the likelihood of a symbol being in error. Using this likelihood and the update, we propose a likelihood ascent search (LAS) algorithm to find an update which is not restricted to be in a fixed neighborhood and seeks to provide maximum reduction in ML cost. This process continues till there is a reduction in the ML cost. Compared to existing LAS based algorithms, it is found to provide better error performance, that too at a lower complexity.**

I. Introduction

EFFICIENT detection in a multiple-input multiple-output (MIMO) system with a large number of antennas is a key challenge [1], [2]. Several detection algorithms [3]–[11] have been proposed in the literature to address this problem. Except a few [9]–[11] all others belong to a class of neighborhood search algorithms, primarily categorized into likelihood ascent search (LAS) [3] and reactive tabu search (RTS) [5] algorithms. The main idea of these algorithms is to begin with an initial guess, generate a neighborhood around the initial guess and replace it with the best solution in the neighborhood (in case of LAS the initial guess is also included). In LAS this process continues till there is no further reduction in the ML cost while in RTS some additional polices are used to avoid cycles and early terminations.

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 to update 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. We derive an expression for the best update for the $\mathcal{L}$ locations which are in error. For determining the erroneous locations, we use a likelihood which is based on the channel matrix and the error vector. The likelihoods are arranged in the decreasing order of their magnitude and the first $\mathcal{L}$ locations are selected. The update for these locations can be found using the derived expression. This is done for all the values of $\mathcal{L}$. Finally we select the update which has the minimum ML cost. The iterations stop when there is no further reduction in the ML cost.

We call the proposed algorithm as an unconstrained LAS (ULAS) algorithm and expect it to perform better in terms of both error performance and complexity. This is because ULAS does not restrict itself to a fixed neighborhood and computes the best update in the entire space. Therefore the problem of early termination leading to loss in error performance or the high number of intermediate iterations leading to high complexity, both can be avoided. We corroborate this for large MIMO systems using extensive simulations.

The rest of this paper is structured as follows: Section II describes the system model and discusses the existing neighborhood search algorithms. Based on the neighborhood definition we formulate an optimization problem in Section III and propose an detection algorithm in Section IV. In Section V, we analyze its complexity. Section VI provides the simulations results and finally Section VII concludes the paper. In this paper bold face capital letters denotes matrices, capital letters denotes vectors and small letters is used to denote a scalar quantity. Bar is used to denote a complex quantity.

II. Preliminaries

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

$$\bar{Y} = \bar{\mathbf{H}}\bar{X} + \bar{N}, \tag{1}$$

where $\bar{Y} = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_{N_r})^T$ in which $\bar{y}_i$ represents data received at the ith receive antenna and $\bar{X} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_{N_t})^T$ 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.

This work was supported by Tata Consultancy Services Limited.

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 $\bar{N} = (\bar{n}_1, \bar{n}_2, \cdots, \bar{n}_{N_r})^T$ 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

$$Y = \mathbf{H}X + N, \tag{2}$$

where $Y = (\Re\{\bar{Y}\}^T \Im\{\bar{Y}\}^T)^T$ is a $(2N_r \times 1)$ real equivalent received vector and $X = (\Re\{\bar{X}\}^T \Im\{\bar{X}\}^T)^T$ represents a $(2N_t \times 1)$ real equivalent transmit vector where each $x_i \in \Omega$. Now, the set $\Omega = \{\pm 1, \pm 3 \cdots \pm (\sqrt{M} - 1)\}$ is a set of $\sqrt{M}$ real symbols drawn from a one dimensional constellation i.e. $\sqrt{M}$-PAM, $N = (\Re\{\bar{N}\}^T \Im\{\bar{N}\}^T)^T$ is a $(2N_r \times 1)$ equivalent noise vector and $\mathbf{H}$ denotes 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 equivalent real system model (2) throughout the paper.

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

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

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

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

and we 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 small number of antenna pairs.

In the literature, for MIMO systems up to hundred antenna pairs, several neighborhood search algorithms such as [3]–[5] have been suggested. These algorithms proceed iteratively, initialize with an initial solution vector and then search for the best solution in the neighborhood of the initial solution vector. The $\mathcal{L}$ symbol neighborhood $\mathcal{N}_{\mathcal{L}}(X)$ of a vector X is defined as follows: Consider a vector $\tilde{X}$ which differs from X at exactly $\mathcal{L}$ symbols then $\tilde{X} \in \mathcal{N}_{\mathcal{L}}(X)$. The set which contains these $\mathcal{L}$ indices is defined as $\mathcal{I}_k$. It can be seen that for an $\mathcal{L}$ symbol neighborhood there will be a total of $\binom{2N_t}{\mathcal{L}}$ $\mathcal{I}_k$s. Mathematically, a neighboring vector $\tilde{X} \in \mathcal{N}_{\mathcal{L}}(X)$ can be expressed as

$$\tilde{x}_i = \begin{cases} \omega_j, & \forall \omega_j \neq x_i,\, i \in \mathcal{I}_k \;\&\; \omega_j \in \Omega \\ x_i, & i \notin \mathcal{I}_k, \end{cases} \tag{6}$$

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

Based on the above neighborhood definition there exist different versions of neighborhood search algorithms such as 1-LAS algorithm [3], MLAS algorithm [4], multiple output selection LAS [8], RTS [5], random restart reactive tabu search [6], layered tabu search [7] etc. Except MLAS, all of these algorithms limit the search within a one symbol neighborhood or in other words update only one element at each iteration. This may lead to either an early termination as in the case of 1-LAS or require a high number of iterations as in the case of RTS which may lead to a high computational complexity. Motivated by this, in the next section we revisit the neighborhood search algorithm and construct an optimization problem for finding an unconstrained $\mathcal{L}$ symbol update.

III. Problem Formulation

Let us denote the initial solution vector at the rth iteration as $X^{(r)}$ and the solution vector after the rth iteration or the initial vector for the $(r+1)$th iteration as the sum of $X^{(r)}$ and $P^{(r)}$ where $P^{(r)}$ is an update vector i.e. $X^{(r+1)} = X^{(r)} + P^{(r)}$. The ML cost of $X^{(r+1)}$ can be expressed as a function of the cost for the previous iteration and the update vector using Taylor series expansion as

$$\begin{aligned} &\phi(X^{(r+1)}) = \phi(X^{(r)} + P^{(r)}) \\ &= \phi(X^{(r)}) + \phi'(X^{(r)})P^{(r)} + \tfrac{1}{2}{P^{(r)}}^T \phi''(X^{(r)}) P^{(r)}. \end{aligned} \tag{7}$$

Third and higher derivatives of the Taylor series expansion go to zero and the non-zero terms are given as

$$\phi'(X^{(r)}) = -2(Y - \mathbf{H}X^{(r)})^T \mathbf{H}, \tag{8}$$

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

Using the above expressions the difference in the cost for two consecutive iterations is given by

$$\begin{aligned} \Delta\phi(P^{(r)}) &= \phi(X^{(r+1)}) - \phi(X^{(r)}) \\ &= -2(Y - \mathbf{H}X^{(r)})^T \mathbf{H}P^{(r)} + {P^{(r)}}^T \mathbf{H}^T\mathbf{H}P^{(r)} \\ &= {P^{(r)}}^T \mathbf{W} P^{(r)} - 2F^{(r)}P^{(r)} \end{aligned} \tag{10}$$

where $\mathbf{W} \triangleq \mathbf{H}^T\mathbf{H}$ and $F^{(r)} \triangleq (Y - \mathbf{H}X^{(r)})^T\mathbf{H}$.

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. $\Delta\phi < 0$ and for a given rth iteration, the update which minimizes $\Delta\phi(P^{(r)})$ or maximizes the reduction in ML cost is the optimal update P^*. In view of this, we wish to explore an update which is not constrained to belong to a specified neighborhood i.e. it can be from any part of the entire space.

We refer to it as an unconstrained update which can be mathematically expressed as

$$\begin{aligned} P^* &= \underset{P^{(r)} \in \vartheta^{2Nt}}{argmin} \quad \Delta\phi(P^{(r)}) \\ &= \underset{P^{(r)} \in \vartheta^{2Nt}}{argmin} \quad P^{(r)^T}\mathbf{W}P^{(r)} - 2F^{(r)}P^{(r)} \end{aligned} \tag{11}$$

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 $P^{(r)}$ can take values from the set ϑ, the size of which is $2\sqrt{M}-1$. Hence there are $(2\sqrt{M}-1)^{2N_t}$ number of 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

$$\Delta\phi = \sum_{i=1}^{2N_t}\sum_{j=1}^{2N_t}\langle H_i, H_j\rangle p_i^{(r)}p_j^{(r)} - 2\sum_{i=1}^{2N_t} f_i^{(r)}p_i^{(r)} \tag{12}$$

where $p_i^{(r)}$ denotes the ith element of $P^{(r)}$, H_i represents the ith column of the channel matrix $\mathbf{H}$ and $\langle A, B\rangle$ denotes the inner product of vector A and B. In the above expression any one $p_i^{(r)} = 0$ will lead to $4N_t$ number of terms to zero and the dimension of the problem reduces by one. Let us assume that there is a need to update only $\mathcal{L}$ symbols. It means for that update (which has $(2N_t - \mathcal{L})$ number of zeros in $P^{(r)}$), we can express the problem in (11) as a reduced dimension problem. For a given value of $\mathcal{L}$ there are $\binom{2N_t}{\mathcal{L}}$ $\mathcal{I}_k$s and each $\mathcal{I}_k$ contains $\mathcal{L}$ indices. For a given $\mathcal{L}$ and $\mathcal{I}_k$, we can express (11) as

$$P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^*} = \underset{P_{(\mathcal{L},\mathcal{I}_k)}^{(r)} \in \vartheta^{\mathcal{L}}}{argmin} P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^T}\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}P_{(\mathcal{L},\mathcal{I}_k)}^{(r)} - 2F_{(\mathcal{L},\mathcal{I}_k)}^{(r)}P_{(\mathcal{L},\mathcal{I}_k)}^{(r)} \tag{13}$$

where $P_{(\mathcal{L},\mathcal{I}_k)}^{(r)}$ is an $\mathcal{L}\times 1$ update vector and $\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}$ is an $\mathcal{L}\times\mathcal{L}$ sub matrix constructed by deleting the $(2N_t - \mathcal{L})$ rows and $(2N_t - \mathcal{L})$ columns of $\mathbf{W}$, the indices of which are not part of $\mathcal{I}_k$. Similarly $F_{(\mathcal{L},\mathcal{I}_k)}^{(r)}$ is an $\mathcal{L}\times 1$ vector, formed by deleting the elements from $F^{(r)}$ the indices of which are not part of $\mathcal{I}_k$. The update which minimizes (13) is the best update $P_{\mathcal{L}}^{(r)}$ at the rth iteration for the given $\mathcal{L}$ and $\mathcal{I}_k$ and we denote it by $P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^*}$.

For finding $P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^*}$ we take the partial derivative approach. However, before proceeding it is necessary to 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 the partial derivative and equating it to zero, we get

$$\nabla\Delta\phi = 2\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}P_{(\mathcal{L},\mathcal{I}_k)}^{(r)} - 2F_{(\mathcal{L},\mathcal{I}_k)}^{(r)^T} = 0. \tag{14}$$

Therefore the best update for a given $\mathcal{L}$ and $\mathcal{I}_k$ will be

$$P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^*} = \left\lceil \mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}^{-1}F_{(\mathcal{L},\mathcal{I}_k)}^{(r)^T} \right\rfloor, \tag{15}$$

where $\lceil\cdot\rfloor$ represents element-wise rounding on the alphabet set ϑ. Interestingly, it may be noted that the result for the best one symbol update in [4] is a special case of (15) for $\mathcal{L} = 1$.

Let us denote the update as

$$P_{(\mathcal{L},\mathcal{I}_k)}^{(r)^*} = \mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}^{-1}F_{(\mathcal{L},\mathcal{I}_k)}^{(r)^T} + \delta_{(\mathcal{L},\mathcal{I}_k)}^{(r)} \tag{16}$$

where $\delta_{(\mathcal{L},\mathcal{I}_k)}^{(r)}$ is the rounding error. Substituting the solution in (15) into (13) we get the reduction in the cost function i.e. $\Delta\phi_{(\mathcal{L},\mathcal{I}_k)}^{(r)}$ as given in (17). The values of $\mathcal{L}$ and $\mathcal{I}_k$ for which the reduction in the cost function is maximized or $\Delta\phi_{(\mathcal{L},\mathcal{I}_k)}^{(r)}$ is minimized can be used to find the optimal update P^*. We express this equivalent optimization problem in (18) and denote the optimal value of $\mathcal{L}$ and $\mathcal{I}_k$ as $\mathcal{L}^*$ and $\mathcal{I}_k^*$ respectively.

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

IV. Proposed Algorithm

We propose an $\mathcal{L}^*$ symbol update based LAS algorithm that will seek to find the update at each iteration which maximizes the reduction in ML cost. The complete algorithm is discussed step by step in the subsequent subsections.

A. Initialization

The algorithm is initialized with a solution vector $X^{(0)}$. We may start with a random guess but similar to other neighborhood search algorithms, the error performance and complexity of the algorithm depends 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 these are given below

$$\begin{aligned} X_{MF}^{(0)} &= \mathbf{H}^H Y, \\ X_{ZF}^{(0)} &= (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H Y, \\ X_{MMSE}^{(0)} &= (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I})^{-1}\mathbf{H}^H Y, \end{aligned} \tag{19}$$

where the subscripts denote the type of the receiver. One can begin with any one of the above initial vectors.

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

We compute $\mathbf{W}$ and $F^{(r)}$ using the relationships: $\mathbf{W} = \mathbf{H}^T\mathbf{H}$ and $F^{(r)} = (Y - \mathbf{H}X^{(r)})^T\mathbf{H}$. It may be noted that $\mathbf{W}$ is independent of iterations and needs to be computed only once while $F^{(r)}$ needs to be computed at each iteration.

$$\Delta\phi^{(r)}_{(\mathcal{L},\mathcal{I}_k)} = \left(\mathbf{W}^{-1}_{(\mathcal{L},\mathcal{I}_k)}F^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)} + \delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\right)^T \mathbf{W}_{(\mathcal{L},\mathcal{I}_k)} \left(\mathbf{W}^{-1}_{(\mathcal{L},\mathcal{I}_k)}F^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)} + \delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\right) - 2F^{(r)}\left(\mathbf{W}^{-1}_{(\mathcal{L},\mathcal{I}_k)}F^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)} + \delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\right)$$

$$= -F^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\mathbf{W}^{-1}_{(\mathcal{L},\mathcal{I}_k)}F^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)} + \delta^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)}\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}\delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}, \tag{17}$$

$$\mathcal{L}^*, \mathcal{I}^*_k = \underset{\mathcal{L},\mathcal{I}_k}{argmin}\left\{-F^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\mathbf{W}^{-1}_{(\mathcal{L},\mathcal{I}_k)}F^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)} + \delta^{(r)^T}_{(\mathcal{L},\mathcal{I}_k)}\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}\delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}\right\}. \tag{18}$$

C. Determining the candidate sequence $\mathcal{T}$

For a given $F^{(r)}$ and $\mathbf{W}$ the optimization problem in (18) is a function of $\mathcal{L}$ and $\mathcal{I}_k$. We would like to determine the $\mathcal{L}$ and $\mathcal{I}_k$ which will cause the maximum reduction in the ML cost. We have already noted that finding an optimal $\mathcal{L}$ and $\mathcal{I}_k$ is computationally expensive. We address this issue by noting that for a well conditioned channel matrix $\mathbf{H}$, the noise amplification is low [12]. Using this, it can be argued that the rounding error $\delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ will be small in (16). Since the second term in (18) involves square of the rounding error (which will be even smaller), it can be ignored for the purpose of this subsection. $\mathbf{W}$ and $\mathbf{W}^{-1}$ are nearly diagonal matrices for a well conditioned $\mathbf{H}$ and hence (18) can be approximated by

$$\underset{\mathcal{L},\mathcal{I}_k}{argmin}\sum_{i\in\mathcal{I}_k} -\frac{{f^{(r)}_i}^2}{w_{ii}} = \underset{\mathcal{L},\mathcal{I}_k}{argmax}\sum_{i\in\mathcal{I}_k}\frac{{f^{(r)}_i}^2}{w_{ii}}. \tag{20}$$

Based on this we propose to use $\frac{{f^{(r)}_i}^2}{w_{ii}}$ as a metric for the likelihood of the locations ($i = 1, 2, \cdots 2N_t$) of the erroneous symbols. It turns out that the proposed metric is same as derived in [13]. We sort the indices in the descending order of the likelihood and denote this sequence by $\mathcal{T}$. Using this candidate sequence it will be easy to compute a best $\mathcal{L}$ and $\mathcal{I}_k$ as discussed next.

D. Arranging $X^{(r)}$, $F^{(r)}$ and $\mathbf{W}$

Now we use the candidate sequence $\mathcal{T}$ to arrange the elements in $X^{(r)}$ in the same order as $\mathcal{T}$. By doing this we arrange the erroneous symbols according to their likelihood and an update on the first $\mathcal{L}$ elements will give the best neighbor in the $\mathcal{L}$ symbol neighborhood. The best neighbor is in the sense that it will cause maximum reduction in ML cost if the neighborhood size is $\mathcal{L}$. Since we have arranged $X^{(r)}$, we need to arrange $F^{(r)}$ and $\mathbf{W}$ in the same order as $X^{(r)}$ or $\mathcal{T}$. For arranging $\mathbf{W}$, we can arrange the columns first and then the rows or the otherway round.

E. Best update $P^{(r)^*}_{(\mathcal{L},\mathcal{I}_k)}$ and rounding error $\delta^{(r)^*}_{(\mathcal{L},\mathcal{I}_k)}$

For a given value of $\mathcal{L}$, we select the first $\mathcal{L}$ elements of $\mathcal{T}$, $X^{(r)}$ and $F^{(r)}$ and denote it as $\mathcal{I}_k$, $X^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ and $F^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ respectively. Similarly we select the $\mathcal{L}\times\mathcal{L}$ matrix formed by the intersection of the first $\mathcal{L}$ rows and columns of $\mathbf{W}$ and denote it as $\mathbf{W}_{(\mathcal{L},\mathcal{I}_k)}$. Now we compute the best update $P^{(r)^*}_{(\mathcal{L},\mathcal{I}_k)}$ for a given $(\mathcal{L},\mathcal{I}_k)$ pair using (15). Corresponding to this update a candidate $(r+1)$th solution vector is given by

$$X^{(r+1)}_{(\mathcal{L},\mathcal{I}_k)} = \lceil X^{(r)}_{(\mathcal{L},\mathcal{I}_k)} + P^{(r)^*}_{(\mathcal{L},\mathcal{I}_k)}\rfloor \tag{21}$$

where $\lceil\cdot\rfloor$ 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 $\delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ is given by

$$\delta^{(r)}_{(\mathcal{L},\mathcal{I}_k)} = X^{(r+1)}_{(\mathcal{L},\mathcal{I}_k)} - X^{(r)}_{(\mathcal{L},\mathcal{I}_k)} - P^{(r)}_{(\mathcal{L},\mathcal{I}_k)}. \tag{22}$$

Since $\mathcal{L}$ can take values from 1 to $2N_t$, we get $2N_t$ candidates solution vector for the $(r+1)$th iteration.

F. Update for the rth iteration and terminaton

Now that we have $2N_t$ possible updates, we need to chose the best for the $(r+1)$th iteration. By calculating the update for all possible value of $\mathcal{L}$ we are effectively computing the best update for every possible size of the neighborhood. We select the update $P^{(r)^*}_{(\mathcal{L},\mathcal{I}_k)}$ for which $\Delta\phi^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ (17) is minimum and denote the corresponding $(\mathcal{L},\mathcal{I}_k)$ as $(\mathcal{L}^*,\mathcal{I}^*_k)$. Since we have pruned the update vector in IV-E to $\mathcal{L}$ elements, we need to append $(2N_t - \mathcal{L}^*)$ zeros in $P^{(r)^*}_{(\mathcal{L}^*,\mathcal{I}^*_k)}$ and then reorder the elements according to $\mathcal{T}$. This rearranged version of $P^{(r)^*}_{(\mathcal{L}^*,\mathcal{I}^*_k)}$ is the best update for the rth iteration and we denote it by $P^{(r)^*}$ which is added to the unsorted $X^{(r)}$ ($X^{(r)}$ before IV-D) and rounded off to the set Ω to arrive at the solution vector for the $(r+1)$th iteration.

We repeat the steps from IV-B to IV-F till there is a reduction in the ML cost, else we terminate the algorithm.

V. Complexity of the Proposed Algorithm

One can observe that the proposed algorithm is iterative. Therefore, it is important to examine the complexity of computing the candidate sequence $\mathcal{T}$, the updates $P^{(r)^*}_{\mathcal{L},\mathcal{I}_k}$ and the corresponding $\Delta\phi^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$. Even among these, finding $P^{(r)^*}_{\mathcal{L},\mathcal{I}_k}$ and $\Delta\phi^{(r)}_{(\mathcal{L},\mathcal{I}_k)}$ are more crucial because they need to be computed multiple times in each iteration. The complexity of finding $\mathcal{T}$ depends on the computations required for determining $F^{(r)}$ and the diagonal elements of $\mathbf{W}$. For $N_t = N_r$, it is of the order of $O(N_t^2)$.

One can notice from (15) and (17) that for the computation of $P^{(r)^*}_{\mathcal{L}}$ and $\Delta\phi^{(r)}_{\mathcal{L}}$ respectively we need to compute $\mathbf{W}^{-1}_{\mathcal{L}}$ (for simplicity here we omit the second subscript $\mathcal{I}_k$). The symmetric structure of $\mathbf{W}_{\mathcal{L}}$ gives us freedom to compute $\mathbf{W}^{-1}_{\mathcal{L}+1}$ as a function of $\mathbf{W}^{-1}_{\mathcal{L}}$. If a symmetric matrix $\mathbf{W}_{\mathcal{L}+1}$ can be expressed in terms of another symmetric matrix $\mathbf{W}_{\mathcal{L}}$ as

$$\mathbf{W}_{\mathcal{L}+1} = \begin{bmatrix} \mathbf{W}_{\mathcal{L}} & W_{1:\mathcal{L},\mathcal{L}+1} \\ W_{\mathcal{L}+1,1:\mathcal{L}} & W_{\mathcal{L}+1,\mathcal{L}+1}\end{bmatrix} \tag{23}$$

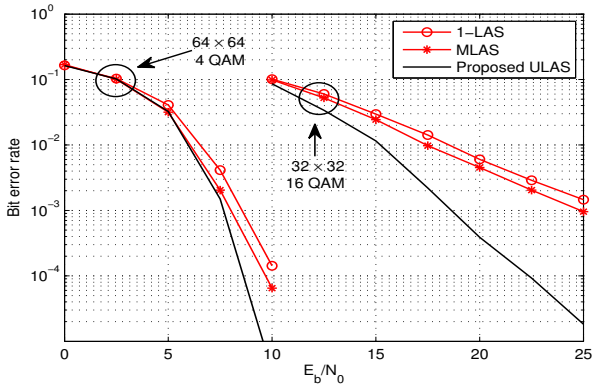

Fig. 1. BER Comparison of the proposed algorithm for a) 32×32 MIMO system with 16-QAM modulation and b) 64×64 MIMO system with 4-QAM modulation.

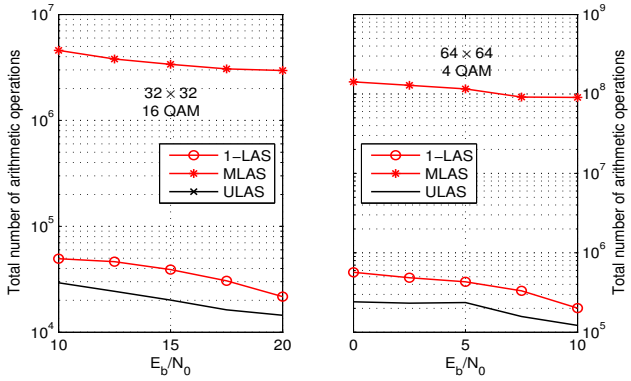

Fig. 2. Number of arithmetic operation required by the proposed algorithm for a) 32×32 MIMO system with 16-QAM modulation and b) 64×64 MIMO system with 4-QAM modulation.

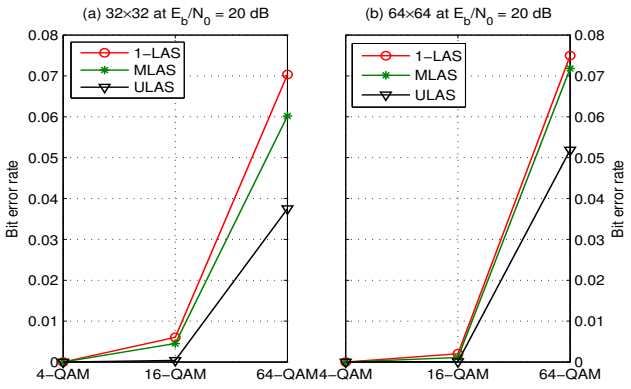

Fig. 3. BER comparison for 32×32 and 64×64 MIMO system with different modulation schemes at $E_b/N_0 = 20$ dB.

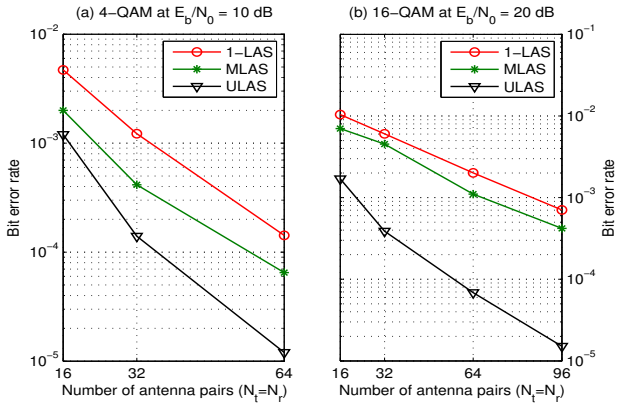

Fig. 4. BER comparison with the increasing number of antennas for (a) 4-QAM modulation at $E_b/N_0 = 10$ dB and (b) 16-QAM modulation at $E_b/N_0 = 20$ dB.

and $\mathbf{W}_{\mathcal{L}}^{-1}$ exists then we know that $\mathbf{W}_{\mathcal{L}+1}^{-1}$ can be written in terms of $\mathbf{W}_{\mathcal{L}}^{-1}$ as follows

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

where $\rho = W_{\mathcal{L}+1,\mathcal{L}+1} - b^T \mathbf{W}_{\mathcal{L}}^{-1} b$ and $b = W_{1:\mathcal{L},\mathcal{L}+1} = W_{\mathcal{L}+1,1:\mathcal{L}}^T$. This has a computational complexity of the order of $O(\mathcal{L}^2)$. Hence the computations required for finding $\mathbf{W}_{\mathcal{L}}^{-1}$ and the corresponding $\Delta\phi_{\mathcal{L}}^{(r)}$ $\forall \mathcal{L} = 1, 2 \cdots, 2N_t$ is of the order of $O(N_t^3)$. Combining all of the steps, the per iteration complexity of the proposed ULAS algorithm is of the order of $O(N_t^3)$.

VI. Simulation Results

In this section, we compare the proposed ULAS algorithm with 1-LAS and MLAS for 32×32 and 64×64 MIMO systems for 4-QAM and 16-QAM modulations respectively. The bit error performance is shown in Fig. 1 while the required number of arithmetic operation (as a measure of complexity) is shown in Fig. 2. From the figures it can be observed that the proposed ULAS not only improves the error performance but also requires a lower number of arithmetic operations compared to both the variants of LAS.

For a 32×32 MIMO system (16-QAM), ULAS requires 7 dB less E_b/N_0 to achieve a BER of 10^{-3} (see Fig. 1) compared to MLAS and nearly 8 dB less compared to 1-LAS. From Fig. 2 we can observe that for the same system at 20 dB the number of arithmetic operations required is 34% less as compared to 1-LAS while compared to MLAS the savings is more than 99%. The trend for E_b/N_0 gain and savings in complexity have been found to be similar for other systems. Thus we can conclude that the proposed ULAS is significantly better in terms of both error performance and 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. 3. 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.

In addition, we have also examined the error performance as

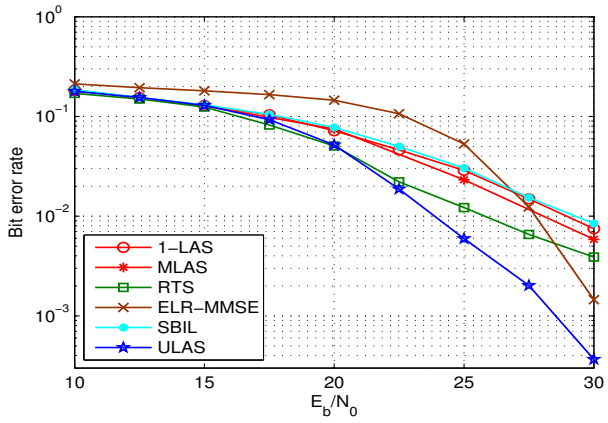

Fig. 5. BER comparison for 64×64 MIMO system with 64-QAM modulation taking MMSE solution as the initial vector.

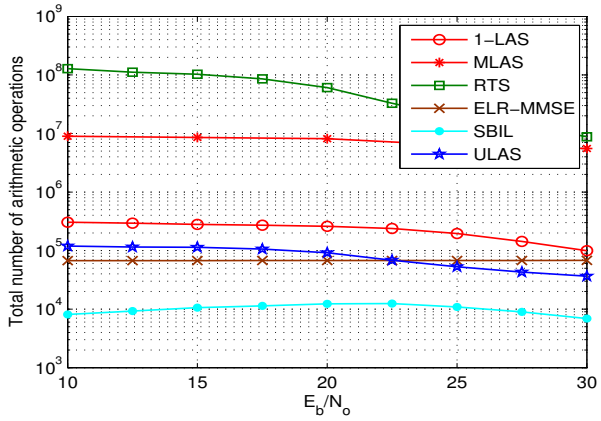

Fig. 6. Number of arithmetic operations for 64×64 MIMO system with 64-QAM modulation taking MMSE solution as the initial vector.

a function of the number of antennas. The simulation results are shown in Fig. 4 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.

Now we compare the performance of the proposed ULAS with other existing detectors, specifically with RTS, element based lattice reduction aided MMSE (ELR-MMSE) detector [9] and sparsity boosted iterative linear detector (SBIL) [11] for a 64×64 MIMO system with 64-QAM modulation. The bit error performance is shown in Fig. 5 while the total number of arithmetic operations required is shown in Fig. 6. From these figures we can observe that the ULAS has a better error performance compared to all the others throughout the range of E_b/N_0. In terms of number of operations, ELR-MMSE is comparable to ULAS while SBIL is better than ULAS. However it may be noted that compared to ELR-MMSE and SBIL, ULAS provides a gain of 4 dB and 6 dB respectively at a BER of 10^{-2}.

VII. Conclusion

An unconstrained update based LAS algorithm has been presented for detection in large MIMO systems. We derived a close form expression for finding the update at the erroneous locations. Since we do not know the erroneous locations we propose a metric for their likelihood. Based on the likelihood we select the $\mathcal{L}$ most likely erroneous locations and then find an update for those symbols. This is done for all the values of $\mathcal{L}$. The update which minimizes the cost function is selected for the next iteration. The process continues till there is an improvement 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 1-LAS and MLAS. It also compares favorably with RTS, ELR-MMSE and SBIL algorithms.

References

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

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

[3] K. Vardhan, S. Mohammed, A. Chockalingam, and B. Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 3, pp. 473–485, April 2008.

[4] S. Mohammed, A. Chockalingam, and B. Sundar Rajan, "A low-complexity near-ML performance achieving algorithm for large MIMO detection," in *Information Theory, ISIT 2008. IEEE International Symposium on*, July 2008, pp. 2012–2016.

[5] B. Rajan, S. Mohammed, A. Chockalingam, and N. Srinidhi, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," in *Information Theory, ISIT 2009. IEEE International Symposium on*, June 2009, pp. 1993–1997.

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

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

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

[9] Q. Zhou and X. Ma, "Element-based lattice reduction algorithms for large MIMO detection," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 2, pp. 274–286, February 2013.

[10] P. Svac, F. Meyer, E. Riegler, and F. Hlawatsch, "Soft-heuristic detectors for large MIMO systems," *Signal Processing, IEEE Transactions on*, vol. 61, no. 18, pp. 4573–4586, Sept 2013.

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

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

[13] A. K. Sah and A. K. Chaturvedi, "Reduced neighborhood search algorithms for low complexity detection in MIMO systems," in *Global Communications, GLOBECOM 2015. IEEE International Symposium on*, Dec 2015. [Online]. Available: http://home.iitk.ac.in/~akc/pdfs/Abhay_GlobeCom_2015.pdf