Low Complexity, Pairwise Layered Tabu Search for Large Scale MIMO Detection

SOURAV CHAKRABORTY (✉ schakraborty.ec@gmail.com)

Indian Institute of Engineering Science and Technology, Shibpur <https://orcid.org/0000-0001-6866-9374>

Nirmalendu Bikas Sinha

Maharaja Nandakumar Mahavidyalaya

Monojit Mitra

Indian Institute of Engineering Science and Technology, Shibpur

Research Article

Keywords: Large scale MIMO, SD, FSD, PLTS, LTS, RTS

Posted Date: October 29th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1003349/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Low Complexity, Pairwise Layered Tabu Search for Large Scale MIMO Detection

Sourav Chakraborty · Nirmalendu Bikas
Sinha · Monojit Mitra.

the date of receipt and acceptance should be inserted later

Abstract This paper presents a low complexity pairwise layered tabu search (PLTS) based detection algorithm for a large-scale multiple-input multiple-output (MIMO) system. The proposed algorithm can compute two layers simultaneously and reduce the effective number of tabu searches. A metric update strategy is developed to reuse the computations from past visited layers. Also, a precomputation technique is adapted to reduce the redundancy in computation within tabu search iterations. Complexity analysis shows that the upper bound of initialization complexity in the proposed algorithm reduces from $O(N_t^4)$ to $O(N_t^3)$. The detection performance of the proposed detector is almost the same as the conventional complex version of LTS for 64QAM and 16QAM modulations. However, the proposed detector outperforms the conventional system for 4QAM modulation, especially in 16×16 and 8×8 MIMO. Simulation results show that the per cent of complexity reduction in the proposed method is approximately 75% for 64×64 , 64QAM and 85% for 64×64 16QAM systems to achieve a BER of 10^{-3} . Moreover, we have proposed a layer-dependent iteration number that can further reduce the upper bound of complexity with minor degradation in detection performance.

Keywords Large scale MIMO · SD · FSD · PLTS · LTS · RTS

S. Chakraborty
IEST Shibpur, West Bengal, India
Tel.: +91-90-46680012
E-mail: schakraborty.ec@gmail.com

N. B. Sinha
Maharaja Nandakumar Mahavidyalaya, West Bengal, India

M. Mitra
IEST Shibpur, West Bengal, India

1 Introduction

Large scale MIMO system at the base station can improve the spectral efficiency [11, 12]. However, the computational complexity for detection increases exponentially with the increase in the number of transmit antennas. The objective of MIMO detection is to attain near maximum likelihood (ML) performance. Sphere decoding (SD) [3] is considered to be the most efficient approach to attain near ML performance. However, its computational complexity grows exponentially with the number of antennas and constellation order, and thus SD is not suitable for the detection of large-scale MIMO systems. Quasi ML detectors such as k-best SD, fixed complexity sphere decoder (FSD), Imbalanced FSD (IFSD) have been proposed in the literature to reduce the computational complexity of SD at the cost of lower detection performance [2, 9]. However, their complexities at higher SNR can be higher than that of the original SD. Therefore, achieving near-optimal detection performance with lower complexity is still a challenging task for large-scale MIMO systems. Various low complexity large-scale/ Massive MIMO detection algorithms have been proposed in the literature. In [19] likelihood ascent search (LAS) and its improved version sequential LAS (SLAS) and Global LAS (GLAS) [15] are proposed for large MIMO systems.

Tabu search (TS) detector is based on local neighborhood searching starting from an initial solution [5, 6, 20]. It has been shown that the computational complexity of TS is far lower than that of SD or FSD for large MIMO systems [18]. In [18, 17] reactive tabu search (RTS) was proposed, and it can obtain near ML performance for BPSK and 4QAM modulations. However, its performance degrades for higher-order modulations like 16QAM and 64QAM. The random restart Tabu search (R3TS) [4] was reported to improve the performance of RTS by running multiple RTS and chooses the best solution vector. Recently developed QR decomposition aided tabu search (QR-TS) [14] reduces the computational load of conventional TS by efficiently utilizing the structure of upper triangular matrix obtained from QR decomposition. A modified QR-TS, namely neighbor-grouped TS (NG-TS) [13] is also proposed to reduce the complexity. However, for higher-order modulations like 64QAM and 8×8 MIMO system, complexity of QR-TS and NG-TS is significantly high, making it unfit for larger antenna systems like 32×32 MIMO with 64-QAM modulation. Layered tabu search (LTS) was proposed further to improve the performance of RTS by layer-wise detection and suitable for large antenna systems [16].

In [7] a modified LTS algorithm for complex-valued channel matrix (LTS-C) [7] was proposed to reduce the complexity of conventional LTS further. LTS-C achieves lower computational complexity as the effective number of detection layers is half of the corresponding real-valued LTS detection algorithm. However, the computational complexity of both LTS and LTS-C are primarily dominated by the initialization step of each RTS. We have optimized the computational complexity of LTS for large MIMO detection in this work. Our contributions in this work are summarized as follows:

- We have proposed a real-valued pairwise layered tabu search algorithm (PLTS), which can utilize the QR matrix efficiently in its computation and computes two layers simultaneously. Thus, the effective number of layer of detection is same as LTC-C.
- An efficient metric update method is proposed between successive layers of tabu searches to reduce the initialization complexity of each tabu search. The proposed metric update method reduces the overall RTS initialization complexity from $O(N_t^4)$ to $O(N_t^3)$.
- The complexity of each tabu search iteration is further optimized by using precomputation technique.
- Layer-dependent iteration is proposed to reduce the number of iterations and complexity. Also, the effect of receive antenna variation is studied.

Outline: The remainder of this paper is organized as follows: Section 2 briefly introduces the system model. Section 3 describes the neighbor definition, the concept of RTS and LTS algorithms. In section 4, proposed LTS and modified RTS algorithms are described. The complexity and BER performance of the proposed and existing algorithms are compared in section 5. Finally, conclusions are drawn in section 6.

Notation: Bold uppercase and lowercase letters represent matrices. The i^{th} column vector of $\mathbf{H}$ is denoted by $\mathbf{H}_i$, and $H_{i,j}$ stands for the element in i^{th} row and j^{th} column of $\mathbf{H}$. The real and imaginary part of a complex number resented by $\Re(\cdot)$ and $\Im(\cdot)$ respectively.

2 Background

2.1 System model

We consider massive multiuser MIMO base station with N_r receive antennas, serving N_t single-antenna users, where $N_t \leq N_r$. The channel is assumed to be frequency flat fading. The received signal vector is given by

$$\tilde{\mathbf{r}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}} \quad (1)$$

Where, $\tilde{\mathbf{r}} = [\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_{N_r}]^T$ is the received signal vector, $\tilde{\mathbf{r}} \in \mathbb{C}^{N_r \times 1}$, $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{N_t}]$ is the transmitted vector. The transmitted symbols $\tilde{x}_i$ are drawn from a complex QAM constellation set $\tilde{\mathbb{A}}$. Thus the transmitted symbol vectors are selected from the set $\tilde{\mathbb{A}}^{N_t}$ consisting of M^{N_t} vectors, i.e. $\tilde{\mathbf{x}} \in \tilde{\mathbb{A}}^{N_t \times 1}$, where M is the number of elements in constellation set $\tilde{\mathbb{A}}$. $\tilde{\mathbf{n}} = [\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_{N_r}]^T$ is an $N_r \times 1$ vector of additive white Gaussian noise (AWGN) samples, whose entris are independent and identically distributed (i.i.d.). $\tilde{\mathbf{H}}$ is the $N_r \times N_t$ channel matrix, where $(i, j)^{th}$ element $\tilde{H}_{ij}$ is the complex channel gain between j^{th} transmit antenna and i^{th} receive antenna, $\tilde{\mathbf{H}} \in \mathbb{C}^{N_r \times N_t}$. The channel state information (CSI) and synchronization is assumed to be perfect at the Base station (BS). The complex system model can be converted to a real system model through real-value decomposition. In this

work we adopt the orthogonal version of real-value decomposition (ORVD) [10] [1] to represent the received signal as:

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \bar{\mathbf{n}} \quad (2)$$

Where, $\mathbf{r} \in \mathbb{R}^{2N_r \times 1}$ and $\bar{\mathbf{n}} \in \mathbb{R}^{2N_r \times 1}$. The elements of $\mathbf{r}$ are defined as $r_{2i-1} = \Re(\tilde{r}_i)$, $r_{2i} = \Im(\tilde{r}_i)$, $i = 1, 2, \dots, N_r$. $\mathbf{x} = [\Re(\tilde{x}_1), \Im(\tilde{x}_1), \dots, \Re(\tilde{x}_{N_t}), \Im(\tilde{x}_{N_t})]^T$ is the real valued equivalent of $\tilde{\mathbf{x}}$. The entries of $\mathbf{x}$ are drawn from the constellation set $\mathbb{A}$ where, $\mathbb{A}$ is real valued signal constellation set of $\tilde{\mathbb{A}}$ and size of $\mathbb{A}$ is $\sqrt{M}$. For example, $\mathbb{A} = \{-1, 1\}$ for QPSK/4QAM and $\mathbb{A} = \{-3, -1, 1, 3\}$ for 16-QAM. $\mathbf{H} \in \mathbb{R}^{2N_r \times 2N_t}$ and the $\mathbf{H}$ defined as

$$\mathbf{H} = \begin{bmatrix} \Re(\tilde{H}_{1,1}) & -\Im(\tilde{H}_{1,1}) & \cdots & \Re(\tilde{H}_{1,N_t}) & -\Im(\tilde{H}_{1,N_t}) \\ \Im(\tilde{H}_{1,1}) & \Re(\tilde{H}_{1,1}) & \cdots & \Im(\tilde{H}_{1,N_t}) & \Re(\tilde{H}_{1,N_t}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Re(\tilde{H}_{N_r,1}) & -\Im(\tilde{H}_{N_r,1}) & \cdots & \Re(\tilde{H}_{N_r,N_t}) & -\Im(\tilde{H}_{N_r,N_t}) \\ \Im(\tilde{H}_{N_r,1}) & \Re(\tilde{H}_{N_r,1}) & \cdots & \Im(\tilde{H}_{N_r,N_t}) & \Re(\tilde{H}_{N_r,N_t}) \end{bmatrix} \quad (3)$$

Thus, the elements in $\mathbf{H}$ defined as $H_{2i-1,2j-1} = \Re(\tilde{H}_{ij})$, $H_{2i-1,2j} = -\Im(\tilde{H}_{ij})$, $H_{2i,2j-1} = \Im(\tilde{H}_{ij})$, $H_{2i,2j} = \Re(\tilde{H}_{ij})$, where $i = 1, 2, \dots, N_r$ and $j = 1, 2, \dots, N_t$. QR decomposition can be applied to decompose the real valued channel matrix as: $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where, $\mathbf{Q} \in \mathbb{R}^{2N_r \times 2N_t}$ is a unitary matrix and $\mathbf{R} \in \mathbb{R}^{2N_t \times 2N_t}$ is an upper triangular matrix. Multiplied by $\mathbf{Q}^T$ both side of (2), the system model can be represented as:

$$\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{v} \quad (4)$$

Where, $\mathbf{y} = \mathbf{Q}^T \mathbf{r}$, $\mathbf{v} = \mathbf{Q}^T \bar{\mathbf{n}}$. The statistics of $\mathbf{v}$ and $\bar{\mathbf{n}}$ are same as $\mathbf{Q}$ is a unitary matrix. The $\mathbf{R}$ matrix obtained from the QR decomposition of the above ORVD channel matrix $\mathbf{H}$ exhibits following properties which will be used in later section to optimize the number of arithmetic operations [10]:

$$R_{2i,2j} = R_{2i-1,2j-1} \quad (5)$$

$$R_{2i-1,2j} = -R_{2i,2j-1} \quad (6)$$

$$R_{2i-1,2i} = 0 \quad (7)$$

Where, $i = 1, 2, \dots, N_t$ and $j = i, i+1, \dots, N_t$. The maximum likelihood solution can be expressed as

$$\begin{aligned} \hat{\mathbf{x}}_{ML} &= \arg \min_{\mathbf{x} \in \mathbb{A}^{2N_t}} \|\mathbf{r} - \mathbf{H}\mathbf{x}\|^2 \\ &= \arg \min_{\mathbf{x} \in \mathbb{A}^{2N_t}} \phi(\mathbf{x}) \end{aligned} \quad (8)$$

Where $\phi(\mathbf{x}) = \mathbf{x}^T \mathbf{H}^T \mathbf{H} \mathbf{x} - 2\mathbf{y}^T \mathbf{H} \mathbf{x}$ is the cost metric of solution vector $\mathbf{x}$. The computation complexity of (8) is exponential with N_t and detection of large scale MIMO system requires low complexity solution. Hence, in this work, we have developed a low complexity solution using layered tabu search.

3 layered Tabu Search

3.1 Neighbor definition

Neighbors of symbol a_q defined as the set $N_G(a_q) \in \mathbb{A} \setminus a_q$. The maximum and minimum number of elements in $N_G(a_q)$ are $\sqrt{M} - 1$ and 1 respectively. We choose N neighbor symbol of a_q based on their Euclidean distance from a_q . For example, $\mathbb{A} = \{-3, -1, 1, 3\}$ for 16-QAM modulation and if we choose $N = 3$, then $N_G(-3) = \{-1, 1, 3\}$, $N_G(-1) = \{-3, 1, 3\}$, $N_G(1) = \{3, -1, -3\}$ and $N_G(3) = \{-3, -1, 1\}$ are all possible symbol in the neighborhood. Let $w_v(a_q)$ denotes v^{th} symbol neighbor of a_q where $v = 1, 2, \dots, N$. Consider an i_M iteration algorithm and $\mathbf{x}^{(m)} = [x_1^{(m)}, x_2^{(m)}, \dots, x_{2N_t}^{(m)}]$ be the solution vector corresponding to the m^{th} iteration where, $x_i^{(m)} \in \mathbb{A}$. Then $(u, v)^{th}$ neighbor vector of $\mathbf{x}^{(m)}$ defined as [18]

$$\mathbf{z}^{(m)}(u, v) = [z_1^{(m)}, z_2^{(m)}, \dots, z_{2N_t}^{(m)}] \quad (9)$$

Where, $u = 1, 2, \dots, 2N_t$ and $v = 1, 2, \dots, N$. The elements of $\mathbf{z}^{(m)}(u, v)$ are selected as

$$z_i^{(m)}(u, v) = \begin{cases} x_i^{(m)} & i \neq u \\ w_v(x_u^{(m)}) & i = u \end{cases} \quad (10)$$

3.2 Reactive tabu search

The algorithm starts from the initial vector $\mathbf{x}^{(0)}$, $\mathbf{g}^{(0)} = \mathbf{x}^{(0)}$ and searches in the neighborhood in i_M iterations. The cost of the initial solution computed as [18]

$$\phi(\mathbf{x}^{(0)}) = \mathbf{x}^{(0)T} \mathbf{W} \mathbf{x}^{(0)} - 2\mathbf{y}^T \mathbf{y}_{MF} \quad (11)$$

Where,

$$\mathbf{W} = \mathbf{H}^T \mathbf{H} \quad (12)$$

$$\mathbf{y}_{MF} = \mathbf{H}^T \mathbf{y} \quad (13)$$

The steps followed in m^{th} iteration are given below:

Step 1: Let

$$\mathbf{f}^{(m)} = \mathbf{W} \mathbf{x}^{(m)} - \mathbf{y}_{MF} \quad (14)$$

$$\mathbf{e} = \mathbf{z}^{(m)} - \mathbf{x}^{(m)} \quad (15)$$

$$\phi(\mathbf{z}^{(m)}(u, v)) = \phi(\mathbf{x}^{(m)}) + C(u, v) \quad (16)$$

Where $\phi(\mathbf{z}^{(m)}(u, v))$ is the cost of $(u, v)^{th}$ neighbor.

$$C(u, v) = e_{u,v} f_u^{(m)} + e_{u,v}^2 W_{u,u} \quad (17)$$

Next, find (u', v') such that

$$(u', v') = \arg \min_{u,v} C(u, v) \quad (18)$$

Check for the acceptance of the move (u', v') using tabu matrix. If this move is accepted, then choose the move as the initial solution of next iteration, i.e $\mathbf{x}^{(m+1)} = \mathbf{z}^{(m)}(u', v')$. If the move is not accepted, then evaluate (18) again excluding the previous unaccepted moves in the same iteration.

Step 2: If $\phi(\mathbf{x}^{(m+1)}) \leq \phi(\mathbf{g}^{(m)})$ then update the minimum cost as $\phi(\mathbf{g}^{(m+1)}) = \phi(\mathbf{x}^{(m+1)})$ and also $\mathbf{g}^{(m+1)} = \mathbf{x}^{(m+1)}$.

Step 3: Update the entris of tabu matrix and also compute $\mathbf{f}^{(m+1)}$ for next iteration as

$$\mathbf{f}^{(m+1)} = \mathbf{f}^{(m)} + (z_u^{(m)}(u, v) - x_u^{(m)})\mathbf{R}_u \quad (19)$$

Stopping condition: The number of iteartions can be reduced by using stoping criteria whenever the algorithm reaches to the ML solution [18].

3.3 Layered Tabu Search (LTS)

This section briely present the concept of LTS in complex domain [16][7]. Assume $\bar{\mathbf{x}} = \mathbf{H}^\dagger \mathbf{r}$, where $\mathbf{H}^\dagger$ is the pseudoinverse of $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{R}}$ is obtained from the QR decomposition of $\tilde{\mathbf{H}}$. Let $\tilde{\mathbf{x}}$ is obtained by rounding each entris in $\bar{\mathbf{x}}$ to the nearest neighbor in $\tilde{\mathbf{A}}$, $\hat{\mathbf{x}}$ be the solution vector, which is determined in N_t steps and initialize $\hat{x}_{N_t} = \tilde{x}_{N_t}$. Then at k^{th} layer, $k = N_t, (N_t - 1), \dots, 1$ following steps are performed:

Step 1:

$$r_k = \bar{x}_k - \frac{1}{\tilde{R}_{k,k}} \sum_{l=k+1}^{N_t} \tilde{R}_{k,l}(\hat{x}_l - \bar{x}_l) \quad (20)$$

Where, $\hat{x}_l$ is l^{th} symbol of $\hat{\mathbf{x}}$, and $\hat{\mathbf{x}}$ is obtained from previous layer.

Step 2: Find the symbol in $\tilde{\mathbf{A}}$, closest to r_k . Let the symbol be $\tilde{a}_q$ and γ is the threshold [16].

- if $|r_k - \tilde{a}_q| < \gamma$, $\hat{x}_k = \tilde{a}_q$, $k = k - 1$ and return to step 1.
- if $|r_k - \tilde{a}_q| \geq \gamma$, set $\hat{x}_k = \tilde{x}_k$. Run RTS with following input

$$\mathbf{x}_{init} = [\hat{x}_k \hat{x}_{k+1}, \dots, \hat{x}_{N_t}] \quad (21)$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{R}_{k,k} & \tilde{R}_{k,k+1} & \dots & \tilde{R}_{k,N_t} \\ 0 & \tilde{R}_{k+1,k+1} & \dots & \tilde{R}_{k+1,N_t} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \tilde{R}_{N_t,N_t} \end{bmatrix} \quad (22)$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}[\bar{x}_k \bar{x}_{k+1} \dots \bar{x}_{N_t}]^T \quad (23)$$

The output of the RTS will be the updated as $\hat{\mathbf{x}} = [\hat{x}_k, \hat{x}_{k+1}, \dots, \hat{x}_{N_t}]$. Make $k = k - 1$ and go back to step 1.

4 Proposed layered tabu search

The dimension of matrices doubles in the real equivalent of a complex-valued system, which doubles the number of layers in LTS. Thus, the chances of RTS and the number of operations increase significantly [7]. The proposed LTS also uses a real-valued system model, but it computes a pair of layers simultaneously. Thus, the proposed PLTS algorithm detects the symbols in N_t steps, where any k^{th} step is termed as ' k^{th} level' and at k^{th} level, $(2k-1)^{th}$ and $2k^{th}$ layers of computations are performed.

4.1 Neighbor difference generation and precomputation

In this section, we shall assume the case where RTS is required to perform at k^{th} level of detection. The condition for RTS at k^{th} level of detection is given in algorithm 1. The dimension of neighbour vectors for each RTS depends on the level of detection. For k^{th} level, $2N(N_t - k + 1)$ neighbor vectors are generated where each vector dimension is $(2(N_t - k + 1) \times 1)$. That is total $4N(N_t - k + 1)^2$ elements required to store for each iteration of k^{th} level. To reduce the amount of storage and efficient cost metric computation, neighbor difference method is proposed. In this case only $e_{u,v} = z_u(u, v) - x_u$ is stored where $z_u(u, v)$ is the u^{th} symbol of $(u, v)^{th}$ neighbor of $\mathbf{x}$ and x_u is u^{th} symbol of $\mathbf{x}$. Thus, the neighbor difference for any symbol $c \in \mathbb{A}$ may be defined as $N_D(c) = N_G(c) - c$, where $N_G(c)$ is the neighbor set for the symbol c . For example, the neighbor difference $N_D(c)$ for real valued equivalent of 16-QAM constellation with $N = 2$ are, $N_D(-3) = \{2, 4\}$, $N_D(-1) = \{-2, 2\}$, $N_D(1) = \{-2, 2\}$ and $N_D(3) = \{-2, -4\}$. The elements of the set N_D are selected from a finite integer set $\mathbb{I}$ and in earlier example, $\mathbb{I} = \{-4, -2, 2, 4\}$. It can be observed from (17) that in each iteration of RTS, $e_{u,v}^2 W_{u,u}$ required to compute for all of $2N(N_t - k + 1)$ neighbors. That is for m iterations, $2mN(N_t - k + 1)$ number of times $e_{u,v}^2 W_{u,u}$ required to compute. To reduce the redundancy in computation, we propose precomputation of $e_{u,v}^2 W_{u,u}$. Thus at the initialization step of RTS, all the possible values of $e_{u,v}^2 W_{u,u}$ are precomputed where $u = 1, 2, \dots, 2(N_t - k + 1)$ and $v = 1, 2, \dots, N$. Therefore, (17) can be modified as

$$C_{u,v} = 2e_{u,v}f_u^{(m)} + \tilde{W}_{u,\delta_{u,v}} \quad (24)$$

Where, $\tilde{W}_{u,\delta_{u,v}} = e_{u,v}^2 W_{u,u}$ values are precomputed at the initialization step of each tabu iteration. The index $\delta_{u,v}$ is the $(u, v)^{th}$ element in square error index matrix $\boldsymbol{\delta}$ and defined as $\delta_{u,v} = \lfloor \frac{e_{u,v}}{d_{min}} \rfloor$, where, d_{min} is the minimum Euclidean distance between any two symbols in alphabet $\mathbb{A}$. Thus, value of $\delta_{u,v}$ are positive integers and $\delta_{u,v} \in \{1, 2, \dots, N\}$. A lookup table may be used to generate N_D and $\boldsymbol{\delta}$ simultaneously as for a certain symbol, both N_D and $\delta_{u,v}$ can be predetermined.

4.2 Efficient cost computation

The earlier section discussed the precomputation technique to reduce redundant computation. In conventional LTS, $\mathbf{W}$ in (12) and $\mathbf{y}_{MF}$ in (13) are computed at the initialization step of every RTS and it increases the number of arithmetic operations significantly. To reduce the number of operations, we propose two methods: i) use the symmetry properties of $\mathbf{W}$ to reduce the number of arithmetic operations and ii) update $\mathbf{W}$ and $\mathbf{y}_{MF}$ obtained from the previous RTS. The $\mathbf{W} = \mathbf{R}^T \mathbf{R}$ matrix exhibits the following properties:

1. $W_{ij} = W_{ji}$ where $i = 1, 2, \dots, 2N_t$ and where $j = 1, 2, \dots, 2N_t$
2. $W_{2i,2i-1} = W_{2i-1,2i} = 0$ where $i = 1, 2, \dots, N_t$
3. $W_{2i-1,2j-1} = W_{2i,2j}$ and $W_{2i-1,2j} = -W_{2i,2j-1}$ where $i = 1, 2, \dots, N_t$ and $j = 1, 2, \dots, N_t$

As $\mathbf{W}$ is a Gram matrix, it follows $W_{ij} = W_{ji}$ and hence no proof is given for property 1. Proof of property 2 and 3 are given in the appendix. Using these above properties it can be written that

$$W_{2i,2j} = W_{2i-1,2j-1} = W_{2j,2i} = W_{2j-1,2i-1} \quad (25)$$

$$W_{2i,2j-1} = -W_{2i-1,2j} = W_{2j-1,2i} = -W_{2j,2i-1} \quad (26)$$

Therefore using the symmetry in $\mathbf{W}$, only N_t^2 elements required to calculate out of $4N_t^2$ elements in $\mathbf{W}$. The $(i, j)^{th}$ element in $\mathbf{W}$ can be computed as

$$\begin{aligned} W_{i,j} &= \sum_{m=1}^i R_{m,i} R_{m,j} \\ &= \sum_{m=1}^2 R_{m,i} R_{m,j} + \dots + \sum_{m=2k-1}^{2k} R_{m,i} R_{m,j} + \dots + \sum_{m=i-1}^i R_{m,i} R_{m,j} \end{aligned} \quad (27)$$

Where, $i = 2, 4, \dots, 2N_t$ and $j = i, i+1, \dots, 2N_t$. Thus, computation of $W_{i,j}$ can be done in terms of i pair of layers starting from i^{th} pair. Lets define the partial computation of $W_{i,j}$ at k^{th} level as:

$$W_{i,j}^{(k)} = \sum_{m=2k-1}^{2k} R_{m,i} R_{m,j} + W_{i,j}^{(k+1)} \quad (28)$$

Where, $k = i, i-1, \dots, 1$ and $W_{i,j}^{(i+1)} = 0$. The final computed value $W_{i,j} = W_{i,j}^{(1)}$. This allows to update $W_{i,j}^{(k)}$ with the help of computations from past layers. Now in case of proposed LTS, upper triangular matrix at k^{th} level expressed as

$$\mathbf{R}^{(k)} = \mathbf{R}_{2k-1:2N_t, 2k-1:2N_t} \quad (29)$$

As, $\mathbf{R}$ shows pair-wise symmetry [10], $\mathbf{R}^{(k)}$ also have identical properties like $\mathbf{R}$ matrix. Also, all the properties defined for $\mathbf{W}$ can be applied for $\mathbf{W}^{(k)} = \mathbf{R}^{(k)T} \mathbf{R}^{(k)}$ at k^{th} level where, dimension of each matrix will be $2(N_t - k +$

$1) \times 2(N_t - k + 1)$. Also, only $(N_t - k + 1)^2$ elements of $\mathbf{W}^{(k)}$ required to update instead of $4(N_t - k + 1)^2$ i.e. only one fourth of the elements required to compute. Now, direct computation of $\mathbf{W}^{(k)}$ increase number of arithmetic operations significantly. Thus, a method is developed to update the elements of $\mathbf{W}^{(k)}$ using the computations from past layers. Lets consider present layer of search is k and the last layer, where RTS was performed $(k+t)$, i.e. $\mathbf{W}^{(k+t)}$ is already available. So, the proposed update method defined as below:

i) The first step is to initialize $\mathbf{W}^{(k)}$ as

$$\mathbf{W}_{2(k+t)-1:2N_t, 2(k+t)-1:2N_t}^{(k)} = \mathbf{W}^{(k+t)} \quad (30)$$

Here, $(k+t) \leq N_t$ and $\mathbf{W}^{(N_t+1)} = 0$. The other elements in $\mathbf{W}^{(k)}$ are set to zero.

ii) Elements of $\mathbf{W}^{(k)}$ can be updated using (28) as

$$W_{i,j}^{(k)} = W_{i,j}^{(k)} + \sum_{m=1}^{2t} R_{m,i}^{(k)} R_{m,j}^{(k)} \quad (31)$$

Where, $i = 2, 4, \dots, 2(N_t - k + 1)$ and $j = i, i + 1, \dots, 2(N_t - k + 1)$ and rest of the elements in $\mathbf{W}^{(k)}$ can be determined using (25) and (26). Similar to the computation of $\mathbf{W}$, the matched filter output $\mathbf{y}_{MF}$ can be computed and update level wise. Now, $\mathbf{y}_{MF}$ in (13) can be determined as

$$\mathbf{y}_{MF} = \mathbf{H}^T \mathbf{r} = \mathbf{R}^T \mathbf{Q}^T \mathbf{r} = \mathbf{R}^T \mathbf{y} \quad (32)$$

The $2i^{th}$ and $(2i-1)^{th}$ element in $\mathbf{y}_{MF}$ can be determined as

$$y_{MF_{2i-1}} = \sum_{m=1}^{2i-1} R_{m,2i-1} y_m = \sum_{m=1}^{2i-2} R_{m,2i-1} y_m + R_{2i-1,2i-1} y_{2i-1} \quad (33)$$

$$y_{MF_{2i}} = \sum_{m=1}^{2i} R_{m,2i} y_m = \sum_{m=1}^{2i-2} R_{m,2i} y_m + R_{2i,2i} y_{2i} \quad (34)$$

Where, $i = 1, 2, \dots, N_t$ and $R_{2i-1,2i} = 0$. Computation of (33) and (34) is equivalent to i levels of operations where at each level, a pair of layers are considered. Now in case of proposed method, computation of $\mathbf{y}_{MF}$ at k^{th} level defined by $\mathbf{y}_{MF}^{(k)} = \mathbf{R}^{(k)T} \mathbf{y}^{(k)}$, where $\mathbf{y}^{(k)} = [y_{2k-1}, y_{2k}, \dots, y_{2N_t}]^T$ and $k = N_t, N_t - 1, \dots, 1$. The i^{th} element in $\mathbf{y}_{MF}^{(k)}$ can be computed as

$$y_{MF_i}^{(k)} = \sum_{m=1}^i R_{m,i}^{(k)} y_m^{(k)} \quad (35)$$

Where, $i \leq 2(N_t - k + 1)$. Direct computation of (35) increase the number of arithmetic operations. Lets assume, $\mathbf{y}_{MF}^{(k+t)}$ is available, i.e. at $(k+t)^{th}$ level the last computation was done and $(k+t) \leq N_t$. Thus, t levels of computation required to update. The proposed low complexity update method is as follows:

- i) Initialize $\mathbf{y}_{MF_{2(k+t):2N_t,1}}^{(k)} = \mathbf{y}_{MF}^{(k+t)}$ and rest of the elements in $\mathbf{y}_{MF}^{(k)}$ set to zero.
- ii) Using (33) and (34), update the i^{th} element as

$$y_{MF_i}^{(k)} = \begin{cases} y_{MF_i}^{(k)} + \sum_{m=1}^{2(t-1)} R_{m,i}^{(k)} y_m^{(k)} + R_{i,i}^{(k)} y_i^{(k)} & i \leq 2t \\ y_{MF_i}^{(k)} + \sum_{m=1}^{2(t-1)} R_{m,i}^{(k)} y_m^{(k)} & i > 2t \end{cases} \quad (36)$$

4.3 Layer dependent iteration

The choice of the iteration number is important for both performance and complexity. A large number of iterations can improve the detection performance significantly but increase the computational complexity. Also, the average number of iterations decreases with an increase in SNR [16]. To reduce the effective number of iterations, we propose layer dependent iteration number. In this method, lower layers assigned a lower value of iteration number. If the minimum and maximum value of iterations are itr_{min} and itr_{max} respectively, then at i^{th} level number of iterations for RTS can be chosen as

$$itr_i = itr_{min} + \left\lceil \frac{(i-1)(itr_{max} - itr_{min})}{N_t} \right\rceil \quad (37)$$

Here, $\lceil x \rceil$ represents the ceiling value of number x .

4.4 Proposed Pairwise LTS algorithm

Algorithm 1 gives the detailed steps of PLTS. The interference from the previously detected layers are cancelled from the present layers (line 6-7). It should be noted that $2k^{th}$ and $(2k-1)^{th}$ layers are computed simultaneously as $R_{2k-1,2k} = 0$. In step 2, the distance between closest symbol and interference cancelled received signal point at $(2k-1)^{th}$ and $2k^{th}$ layers are compared individually with threshold γ (line 10). If both distances are below the threshold, RTS is skipped. For consecutive layer of RTS, t is set to 1 and for $t > 1$, $(t-1)$ is the number of levels at which RTS was skipped. Also, $\mathbf{W}^{(k)}$ and $\mathbf{y}_{MF}^{(k)}$ are updated from the initial value $\mathbf{W}^{(k+t)}$ and $\mathbf{y}_{MF}^{(k+t)}$. After RTS is performed, the partial computation of $\mathbf{W}^{(k)}$ and $\mathbf{y}_{MF}^{(k)}$ at k^{th} level are stored in $\mathbf{W}$ and $\mathbf{y}_{MF}$ respectively for next layers. If RTS is skipped, t will be incremented by one.

Algorithm 1 Proposed pairwise LTS algorithm

```

1: Input:  $Q, R, y, \gamma$ 
2: Output:  $\hat{x}$ 
3:  $\bar{y} = Q^T y, \bar{x} = R^{-1} \bar{y}, \tilde{x} = \text{round}(\bar{x}) \in \mathbb{A}, t = 0$ 
4: for  $k = N_t$  to 1 do
5:   Step 1: Cancel interference from previous layers
6:    $r_{2k-1} = \bar{x}_{2k-1} - \frac{1}{R_{2k-1,2k-1}} \sum_{n=2k+1}^{2N_t} R_{2k-1,n} (\hat{x}_n - \bar{x}_n)$ 
7:    $r_{2k} = \bar{x}_{2k} - \frac{1}{R_{2k,2k}} \sum_{n=2k+1}^{2N_t} R_{2k,n} (\hat{x}_n - \bar{x}_n)$ 
8:   Step 2: Find the closest symbol
9:   Let the closest symbols near  $r_{2k-1}$  and  $r_{2k}$  are  $a_{2k-1}$  and  $a_{2k}$  respectively
10:  if  $|r_{2k-1} - a_{2k-1}| < \gamma$  and  $|r_{2k} - a_{2k}| < \gamma$  then
11:     $\hat{x}_{2k-1} = a_{2k-1}, \hat{x}_{2k} = a_{2k}$ 
12:     $t = t + 1$ 
13:  else
14:     $\hat{x}_{2k-1} = \tilde{x}_{2k-1}, \hat{x}_{2k} = \tilde{x}_{2k}$ 
15:     $\hat{x}^{(k)} = [\hat{x}_{2k-1}, \hat{x}_{2k}, \dots, \hat{x}_{2N_t-1}, \hat{x}_{2N_t}]$ 
16:     $\bar{y}^{(k)} = [\bar{y}_{2k-1}, \bar{y}_{2k}, \dots, \bar{y}_{2N_t-1}, \bar{y}_{2N_t}]$ 
17:     $\hat{R}^{(k)} = R_{2k-1:2N_t, 2k-1:2N_t}$ 
18:     $W^{(k)} = W_{2k-1:2N_t, 2k-1:2N_t}$ 
19:     $y_{MF}^{(k)} = y_{MF_{2k-1:2N_t}}$ 
20:     $n_t = 2(N_t - k + 1)$ 
21:    Update  $W^{(k)}$  and  $y_{MF}^{(k)}$  using (31) and (36) respectively
22:     $[\hat{x}_{2k-1:2N_t}] \leftarrow RTS(y_{MF}^{(k)}, W^{(k)}, \hat{x}^{(k)}, n_t, itr)$ 
23:     $W_{2k-1:2N_t, 2k-1:2N_t} = W^{(k)}$ 
24:     $y_{MF_{2k-1:2N_t}} = y_{MF}^{(k)}$ 
25:     $t = 1$ 
26:  end if
27: end for

```

4.5 Modified low complexity RTS algorithm

Algorithm 2 describes the modified low complexity RTS algorithm. First, pre-computation is performed (line-6) to reuse the values over multiple iterations and cost factor $C_{u,v}$ are computed for all neighbors (line 11). In step 2, acceptance of the move (u', v') is checked using its cost and tabu value (line 14-18). If the move is not accepted, then go back to the line 12, exclude the previous (u', v') move from search and get new (u', v') . If all the moves are rejected, decrease the non-zero entries of the tabu matrix by its minimum non-zero value and go back to step 2. After a move is accepted, its repetition is checked by comparing its cost with the previous iteration selected moves cost. If repetition is found, length of repetition l_{rep} , tabu period P and number of repetition rep_{count} is updated (line 28). Next, cost of accepted move $\phi_{u',v'}$ is compared with the minimum cost $\phi(g^{(m)})$. If accepted move cost is higher then (u', v') move is added to tabu list and also its tabu value is assigned. Else, $\phi(g^{(m+1)})$ is updated with $\phi_{u',v'}$.

Algorithm 2 Modified RTS algorithm

```

1: Input:  $y_{MF}, W, x_{init}, n_t, itr_{max}$ 
2: Output:  $x_{ML}$ 
3: Initialization:  $m = 1, terminate = 0$ 
4:  $W' = Wx$  ;  $\phi(x) = x^T W' - 2y_{MF}^T x$ 
5:  $f^{(1)} = W' - y_{MF}$ 
6: Precompute  $\tilde{W}_{u,\delta_{u,v}} = e_u^2 W_{u,u}$  for all possible values of  $e_u^2$ 
7: while ( $terminate \neq 1$ ) do
8:   Step 1: Finding the best move
9:    $l_{flag} = 0, k = 0$ 
10:  Determine the neighbor difference matrix  $e$  and square error index matrix  $\delta$ 
11:   $C_{u,v} = 2e_{u,v}f_u^{(m)} + \tilde{W}_{u,\delta_{u,v}}$  Where  $u = 1, 2, \dots, n_t$  and  $v = 1, 2, \dots, N$ 
12:  Step 2: Find the minimum value of  $C$  and corresponding index  $(u', v')$ 
13:   $\phi_{u',v'} = \phi(x^{(m)}) + C_{u',v'}$ 
14:  if ( $\phi_{u,v} \leq \phi(g^{(m)})$ ) || ( $T((u-1)\sqrt{M} + q, v) == 0$ ) then
15:     $x^{(m+1)} = x^{(m)}$ 
16:     $x_u^{(m+1)} = x_u^{(m)} + e_{u_1,v_1}$ 
17:     $\phi(x^{(m+1)}) = \phi_{u,v}$ 
18:     $MoveAccepted = 1$ 
19:  else
20:    Exclude  $C(u', v')$  and go back to line 13 and search for the minimum value again
21:  end if
22:  if  $MoveAccepted == 0$  then
23:    Find the minimum nonzero tabu value and decrease the entris of tabu matrix by that value
24:  end if
25:  Step 3: Update tabu value
26:  Find the previous iteration number  $k$  in which this  $\phi(x^{(m+1)})$  solution was found
27:  if ( $k$ ) then
28:     $L_{rep} = m - k, P = P + 1, rep_{count} = rep_{count} + 1, updateP = k$ 
29:  end if
30:  if  $updateP \&\& (m - updateP) > \beta L_{rep}$  then
31:     $P = \max(P - 1, 1)$ 
32:  end if
33:  if  $\phi_{u',v'} \leq \phi(g^{(m)})$  then
34:     $\phi(g^{(m+1)}) = \phi_{u',v'}$ 
35:     $g^{(m+1)} = x^{(m+1)}$ 
36:     $T((u' - 1)\sqrt{M} + q', v') = T((u' - 1)\sqrt{M} + q'', v'') = 0$ 
37:  else
38:     $\phi(g^{(m+1)}) = \phi(g^{(m)})$ 
39:     $g^{(m+1)} = g^{(m)}$ 
40:     $T((u' - 1)\sqrt{M} + q', v') = T((u' - 1)\sqrt{M} + q'', v'') = P + 1$ 
41:     $l_{flag} = 1$ 
42:  end if
43:  Step 4: stopping condition
44:  if ( $m == itr_{max}$ ) || ( $(rep_{count} \geq rep_{max}) \&\& (l_{flag} = 1)$ ) then
45:     $terminate = 1$ 
46:  else
47:    Step 5: Update tabu matrix and  $f^{(m+1)}$ 
48:     $T_{r,s} = \max(T_{r,s} - 1, 0)$  for  $r = 1, 2, \dots, n_t$  and  $s = 1, 2, \dots, N$ 
49:     $f^{(m+1)} = f^{(m)} + e_{u',v'} W_{u'}$ 
50:     $m = m + 1$ 
51:  end if
52: end while
53:  $x_{ML} = g^{(m+1)}$ 

```

The variables q', q'' and v'' (line 36 and 40) are defined by $a_{q'} = x_{u'}^{(m)} = w_{v''}(x_{u'}^{(m+1)})$, $a_{q''} = x_{u'}^{(m+1)}$ and $a_{q'}, a_{q''} \in \mathbb{A}$. Here, $l_{flag} = 1$ indicates that the algorithm reaches to local minima. Step 4 of the algorithm defines the algorithm termination condition. Finally, in step 5, entries of tabu matrix are updated, and $\mathbf{f}^{(m+1)}$ is computed for the next iteration if the termination condition failed.

4.6 Complexity analysis

In this section, we shall examine the amount of complexity reduction that can be achieved by the proposed method.

4.6.1 Complexity reduction in metric update

From (27) it can be observed that computation of $W_{i,j}$ requires i multiplications and $(i-1)$ additions where $i = 2, 4, \dots, 2N_t$ and $j = i, i+1, \dots, 2N_t$. Thus, total number of additions and multiplications are required at i^{th} layer are $(i-1)(2N_t-i+1)$ and $i(2N_t-i+1)$ respectively. The maximum number of operations required when tabu search performed at last level corresponding to $k = 1$ and it is independent of the number of tabu search levels. This is because the proposed metric update method restricts all the redundant computations. Now, consider $i = 2l$ where, $l = 1, 2, \dots, N_t$. Thus, maximum number of multiplications and additions required for all $\mathbf{W}^{(k)}$ computation for $k = N_t, N_t-1, \dots, 1$ can be expressed as:

$$mult_{wp} = \sum_{l=1}^{N_t} 2l(2N_t - 2l + 1) = \frac{1}{3}(2N_t^3 + 3N_t^2 + 1) \quad (38)$$

$$add_{wp} = \sum_{l=1}^{N_t} (2l-1)(2N_t - 2l + 1) = \frac{1}{3}(2N_t^3 + 1) \quad (39)$$

Similarly we can find the number of arithmetic operations for all $\mathbf{y}_{MF}^{(k)}$ in proposed update method. In this case number of arithmetic operations are equal to the number of operations required for multiplication of $\mathbf{R} \in \mathbb{R}^{2N_t \times 2N_t}$ and $\mathbf{y} \in \mathbb{R}^{2N_t \times 1}$. Thus total number of additions and multiplication required are $N_t(2N_t-1)$ and $N_t(2N_t+1)$ respectively.

On the other hand, in conventional approach, $\mathbf{W}^{(k)}$ in (12) and $\mathbf{y}_{MF}^{(k)}$ in (13) are computed at the initialization phase of each RTS and does not utilize any past layer results. Lets consider $n = N_t - k + 1$ as the number of layers associated with k^{th} level and as $k = N_t, N_t-1, \dots, 1$, $n = 1, 2, \dots, N_t$. The total number of additions and multiplications at k^{th} level are $4n^2(2n-1)$ and $8n^3$ for $\mathbf{W}^{(k)}$ and $2n^2$ and $n(2n+1)$ for $\mathbf{y}_{MF}^{(k)}$ respectively. The maximum number of operations requires when tabu search occurs for all the layers. Thus,

the maximum number of additions and multiplications required to compute all $\mathbf{W}^{(k)}$ can be expressed as

$$add_w = \sum_{n=1}^{N_t} (8n^3 - 4n^2) = 2N_t^4 + \frac{2}{3}N_t^3 - 2N_t^2 - \frac{4N_t}{3} \quad (40)$$

$$mult_w = \sum_{n=1}^{N_t} 8n^3 = 2N_t^4 + 4N_t^3 + 2N_t^2 \quad (41)$$

Similarly, the maximum number of additions and multiplications to compute all $\mathbf{y}_{MF}^{(k)}$ can be expressed as

$$add_{ymf} = \sum_{n=1}^{N_t} 2n^2 = \frac{N_t}{3}(N_t + 1)(2N_t + 1) \quad (42)$$

$$mult_{ymf} = \sum_{n=1}^{N_t} (2n^2 + n) = \frac{N_t}{3}(N_t + 1)(2N_t + 1) + \frac{N_t}{2}(N_t + 1) \quad (43)$$

Thus, in proposed update method, the complexity is reduced from $O(N_t^4)$ to $O(N_t^3)$ for $\mathbf{W}$ and $O(N_t^3)$ to $O(N_t^2)$ for $\mathbf{y}_{MF}$. Table 1 shows both algorithms' total additions and multiplications required for RTS initialization in the worst-case scenario.

Table 1: Number of additions and multiplications required for initializing RTS in the worst-case scenario

Operations	Conventional LTS	Proposed PLTS
Additions	$2N_t^4 + \frac{4}{3}N_t^3 - N_t^2 - N_t$	$\frac{2}{3}N_t^3 + 2N_t^2 - N_t$
Multiplications	$2N_t^4 + \frac{14}{3}N_t^3 + \frac{7}{3}N_t^2 + N_t$	$\frac{2}{3}N_t^3 + 3N_t^2 + N_t$

4.6.2 Complexity reduction in precomputation

For each RTS performed at k^{th} level, all the possible values of $e_{u,v}^2 W_{u,u}$ are precomputed. It is already mentioned that the number of possible values of $e_{u,v}^2$ are equal to N . Thus, using property 3 of $\mathbf{W}$, at k^{th} level only nN multiplications are required to precompute all possible $e_{u,v}^2 W_{u,u}$ at the initialization phase of RTS (line 6 of algorithm 2). For all N_t levels of tabu search, total number of multiplications required for precomputation are

$$mult_p = \sum_{n=1}^{N_t} nN = NN_t \frac{(N_t + 1)}{2} \quad (44)$$

In the conventional approach, $e_{u,v}^2 W_{u,u}$ required to compute at every iteration as given in (17). Thus, for each iteration of RTS performed at k^{th} level, $2Nn$ multiplications are required. Suppose total i_n iterations are performed at n^{th} level, then a total $2i_n n N$ number of multiplications are required. Thus precomputation reduces the number of multiplications operations to compute $e_u^2 W_{u,u}$ at k^{th} level by a factor of $2i_n$. It should be noted that i_n is not constant for any level, and it is dependent on the termination condition, and its maximum value is defined at the initialization step. If we assume that tabu search performed at all levels and the number of iterations for each RTS is fixed to i_M , then the total multiplications in the conventional method.

$$mult_c = \sum_{n=1}^{N_t} 2i_M n N = 2i_M N N_t \frac{N_t + 1}{2} = 2i_M mult_p \quad (45)$$

Thus, precomputations can reduce the number of multiplications in conventional LTS upto $2i_M$ times.

5 Performance evaluation

This section evaluated the bit error rate (BER) performance and complexity of the proposed system compared with the conventional complex version of LTS algorithm. Following parameters are used for both conventional LTS and proposed PLTS algorithms: $rep_{max} = 10$, $itr_{max} = 20$, $\beta = 10$, $N = 1$ for 4-QAM, $rep_{max} = 10$, $itr_{max} = 100$, $\beta = 100$ and $N = 3$ for 16-QAM and $rep_{max} = 20$, $itr_{max} = 200$, $\beta = 200$, $N = 2$ for 64-QAM and $P_0 = 1$. The threshold value γ is set to $d_{min}/5$ for our simulation. For a fair comparison, in the simulations, we used sorted QR decomposition (SQRD)[8] based channel matrix ordering for both the detection algorithms. The proposed system entirely uses a real-valued equivalent system. Thus, real-valued SQRD is used for it. The complex model is assumed for conventional LTS, so complex SQRD used for it.

Fig. 1 and 2 shows the comparison between the BER performance of the proposed LTS, conventional LTS, RTS and 1-LAS algorithms with 4-QAM, 16-QAM and 64-QAM modulations for different MIMO systems. The performances of LTS, PLTS, 1-LAS and RTS are almost identical for 64×64 and 32×32 MIMO systems with 4QAM modulation. However, the performance gap is noticeable for 16QAM and 64QAM modulations. In both cases, PLTS and LTS outperform RTS and 1-LAS. On the other hand, proposed PLTS and conventional LTS performs almost identically for higher-order modulation. For 4QAM and 16QAM modulations, the proposed algorithm performs better than conventional LTS in the case of 16×16 and 8×8 MIMO systems.

Fig. 1. BER comparison between the proposed LTS algorithm and conventional complex LTS algorithm with 4-QAM, 16-QAM and 64-QAM modulations (a) 64×64 MIMO systems (b) 32×32 MIMO systems

Fig. 2. BER comparison between the proposed LTS algorithm and conventional complex LTS algorithm with 4-QAM, 16-QAM and 64-QAM modulations (a) 16×16 MIMO systems (b) 8×8 MIMO systems

Fig. 3 and 4 shows different cases of the average number of real operations needed by PLTS and LTS for 64×64 , 32×32 , 16×16 and 8×8 MIMO

antenna systems with 64-QAM, 16-QAM and 4-QAM modulations. It should be noted that the number of arithmetic operations for QR decomposition is not considered here as it is common to both systems and a part of channel matrix preprocessing. The complexity is determined in terms of the total number of real arithmetic operations. The computations performed in the proposed system are all in the real domain. On the other hand, conventional systems use complex arithmetic operations for interference cancellation, and RTS is performed in a real equivalent system.

Fig. 3. Average number of real operations. (a) 64×64 MIMO system, (b) 32×32 MIMO system

It can be observed that the lower limit of each detector's arithmetic complexity converges towards a fixed value. For a specific antenna configuration, these lower limits are close to each other. This happens as at higher SNR, most RTS operations are skipped, and the arithmetic operations are required only for interference cancellations. On the other hand, the upper limit of arithmetic complexity depends on the antenna configuration and modulation order; i.e., as the number of antennas increases, more RTS is required to perform at low SNR. Thus, when most of the layers are required to perform RTS, the maximum number of arithmetic operations are required, which explains the upper limit of complexity shown in plots in Fig. 3 and 4.

Fig. 5 shows the corresponding per cent of complexity reduction achieved in the proposed system. It can be observed that the per cent of complexity reduction is more for higher-order antenna systems. Also, the per cent of complexity reduction increases with SNR and saturates at higher SNR except for 4-QAM systems.

Fig. 4. Average number of real operations. (a) 16×16 MIMO system, (b) 8×8 MIMO system

Fig. 5. Percent of complexity reduction in proposed system for different modulations and antenna orders (a) 64-QAM, (b) 16-QAM, (c) 4-QAM

Table 2 shows the per cent of complexity reduction and corresponding SNR values in the proposed system to achieve a BER of 10^{-3} . The proposed algorithm per cent of complexity reduction depends on the number of transmit antennas and modulation order. The complexity reduction is more for a larger antenna system for a particular modulation. Also, the lower-order modulation schemes can achieve more complexity reduction. The reduction of complexity is achieved mainly due to the efficient computation of matrices and pre-computation techniques. If we consider conventional LTS in a real do-

main instead of a complex domain, complexity reduction would be more, and this happens as the number of layers in real domains doubles. In contrast, the proposed method considers a pair of layers simultaneously, and the number of allowed levels are the same as complex LTS.

Table 2: Percent of complexity reduction and corresponding SNR to achieve a BER of 10^{-3}

	64QAM		16QAM		4QAM	
Antenna order	SNR (dB)	% of reduction	SNR (dB)	% of reduction	SNR (dB)	% of reduction
64×64	30.37	75.37	20.07	85.49	10.88	96.57
32×32	28.97	53.73	20.14	67.91	11.43	91.64
16×16	28.64	39.52	20.21	48.78	11.77	80.16
8×8	28.64	30.64	20.21	35.29	11.77	58.77

Table 3: Per-symbol-complexity (PSC) $\times 10^3$ number of real operations required to achieve a BER of 10^{-3}

	64QAM		16QAM		4QAM	
Antenna order	PLTS PSC	LTS PSC	PLTS PSC	LTS PSC	PLTS PSC	LTS PSC
64×64	18.527	75.296	42.226	291.762	15.047	450.619
32×32	18.067	38.518	19.223	59.865	4.873	59.865
16×16	10.568	17.221	9.200	17.888	1.879	9.549
8×8	6.458	9.210	4.566	7.032	0.908	2.203

Table 3 presents a comparison of the complexity in terms of per-symbol-complexity (PSC), which is defined as the average number of real operations per symbol ($\times 10^3$) required to achieve a specific BER (in our case, it is 10^{-3}). The PSC value for both algorithms reduces with the decrease in the number of transmit antennas, and the proposed PLTS algorithm outperforms the LTS algorithms in all the cases. It can be observed that for 64×64 MIMO system, the PSC value of LTS increases with a decrease in constellation size. This happens as more layers are required to perform RTS, and the initialization step of RTS primarily dominates complexity. However, initialization complexity of the proposed algorithm are not affected by the number of layer visit. Thus, its PSC value is primarily dominated by neighbourhood search operations.

Fig. 6 shows the effect of level-dependent iteration on BER performance and the average number of operations with SNR. The BER performance degrades slightly with the decrease in itr_{min} . However, the average number of arithmetic operations also decreases with the decrease in itr_{min} . Although the floor value of complexity converges to a fixed value for higher SNR, the plot's transition slope is almost the same for all the iteration numbers. The upper bound of arithmetic complexity reduces with itr_{min} as shown in the plot in

Fig. 6(b). A performance loss of 0.5 dB can be seen from Fig. 6(a) to achieve a BER of 10^{-3} when we reduce the itr_{min} from 200 to 120.

Fig. 6. Effect of iteration variation with SNR (a) BER performance (b) Average number of real operations

Fig. 7. Effect of receive antenna variation (a) BER performance, (b) Average number of real operations

Fig. 7 shows the performance of the proposed detector with the increase in the number of receive antennas for a fixed number of transmit antennas $N_t = 32$. The BER performance of the detector improves significantly with

the increase in the number of receive antennas, and the complexity plot shows an increase in the slope of transition with the number of receive antennas. It is worth mentioning that the conventional LTS also shows similar performance improvement with an increase in the number of receiving antennas. The upper and lower bound of the arithmetic complexity are not significantly affected by the smaller variation of the number of receive antennas.

6 Conclusion

We have proposed a low-complexity pair-wise layered tabu search method for a large-scale MIMO system in this work. The proposed algorithm utilizes the metric update method between successive layers of tabu search to reduce the initialization complexity of RTS. The pre-computation technique is developed to further reduce the redundant computation within RTS. The proposed PLTS algorithm shows significant complexity reduction with almost the same detection performance. In addition, level-dependent iteration can further reduce the upper bound of arithmetic complexity with almost identical BER performance. Moreover, an increase in the number of received antennas shows significant performance improvement without affecting the upper and lower bound of complexity much, making it suitable for a massive MIMO system with the number of transmitting to receive antenna ratios close to one.

A Proof of property 2 and 3 of $\mathbf{W}$

Proof of property 2: $W_{2i,2i-1} = W_{2i-1,2i} = 0$

Using (27), $(2i, 2i - 1)^{th}$ element of $\mathbf{W}$ can be expressed as

$$\begin{aligned} W_{2i,2i-1} &= \sum_{m=1}^{2N_t} R_{m,2i} R_{m,2i-1} \\ &= \sum_{n=1}^{N_t} \{R_{2n,2i} R_{2n,2i-1} + R_{2n-1,2i} R_{2n-1,2i-1}\} \quad \text{Where, } m = 2n \end{aligned} \quad (46)$$

From (5) and (6), we can write, $R_{2n,2i} = R_{2n-1,2i-1}$ and $R_{2n,2i-1} = -R_{2n-1,2i}$. Substituting $R_{2n,2i}$ and $R_{2n,2i-1}$ in (46) we can obtain $W_{2i,2i-1} = 0$. Also, $W_{2i-1,2i} = 0$ as $W_{i,j} = W_{j,i}$ from property 1 of $\mathbf{W}$.

Proof of property 3: $W_{2i-1,2j-1} = W_{2i,2j}$ and $W_{2i-1,2j} = -W_{2i,2j-1}$

From (27), $W_{2i-1,2j-1}$ can be computed as

$$\begin{aligned} W_{2i-1,2j-1} &= \sum_{m=1}^{2N_t} R_{m,2i-1} R_{m,2j-1} \\ &= \sum_{n=1}^{N_t} \{R_{2n,2i-1} R_{2n,2j-1} + R_{2n-1,2i-1} R_{2n-1,2j-1}\} \quad \text{Where, } m = 2n \end{aligned} \quad (47)$$

Using (5) and (6), we can substitute $R_{2n,2i-1}$, $R_{2n,2j-1}$, $R_{2n-1,2i-1}$ and $R_{2n-1,2j-1}$ in (47) by $-R_{2n-1,2i}$, $-R_{2n-1,2j}$, $R_{2n,2i}$ and $R_{2n,2j}$ respectively. Therefore, (47) can be

expressed as

$$\begin{aligned} W_{2i-1,2j-1} &= \sum_{n=1}^{N_t} \{R_{2n,2i-1}R_{2n,2j-1} + R_{2n-1,2i-1}R_{2n-1,2j-1}\} \\ &= W_{2i,2j} \end{aligned} \quad (48)$$

In a similar way we can prove $W_{2i-1,2j} = -W_{2i,2j-1}$. The element $W_{2i-1,2j}$ can be expressed as

$$\begin{aligned} W_{2i-1,2j} &= \sum_{m=1}^{2N_t} R_{m,2i-1}R_{m,2j} \\ &= \sum_{n=1}^{N_t} \{R_{2n,2i-1}R_{2n,2j} + R_{2n-1,2i-1}R_{2n-1,2j}\} \quad \text{Where, } m = 2n \end{aligned} \quad (49)$$

Using (5) and (6), we can substitute $R_{2n,2i-1}$, $R_{2n,2j-1}$, $R_{2n-1,2i-1}$ and $R_{2n-1,2j-1}$ in (49) by $-R_{2n-1,2i}$, $-R_{2n-1,2j}$, $R_{2n,2i}$ and $R_{2n,2j}$ respectively. Therefore, (49) can be expressed as

$$\begin{aligned} W_{2i-1,2j} &= - \sum_{n=1}^{N_t} \{R_{2n,2i}R_{2n,2j-1} + R_{2n-1,2i}R_{2n-1,2j-1}\} \\ &= -W_{2i,2j-1} \end{aligned} \quad (50)$$

Conflict of interest

All authors declares that they have no conflict of interest

Funding

None

Availability of data and material

Not applicable

Code availability

yes

Authors' contributions

All the authors have contributed to complete this work and preparing this manuscript.

References

1. Azzam, L., Ayanoglu, E.: Reduced complexity sphere decoding for square QAM via a new lattice representation. GLOBECOM - IEEE Global Telecommunications Conference pp. 4242–4246 (2007)
2. Barbero, L.G., Thompson, J.S.: Fixing the complexity of the sphere decoder for MIMO detection. IEEE Transactions on Wireless Communications **7**(6), 2131–2142 (2008)
3. Burg, A., Borgmann, M., Wenk, M., Zellweger, M., Fichtner, W., Bölcskei, H.: VLSI Implementation of MIMO detection using the sphere decoding algorithm. IEEE Journal of Solid-State Circuits **40**(7), 1566–1576 (2005)
4. Datta, T., Srinidhi, N., Chockalingam, A., Rajan, B.S.: Random-restart reactive tabu search algorithm for detection in large-MIMO systems. IEEE Communications Letters **14**(12), 1107–1109 (2010)
5. Glover, F.: Tabu Search—Part I. ORSA Journal on Computing **1**(3), 190–206 (1989)

6. Glover, F.: Tabu Search—Part II. *ORSA Journal on Computing* **2**(1), 4–32 (1990)
7. Karthikeyan, M., Saraswady, D.: Low complexity layered tabu search detection in large MIMO systems. *AEU - International Journal of Electronics and Communications* **83**, 106–113 (2018)
8. Kim, T.H.: Low-complexity sorted qr decomposition for mimo systems based on pairwise column symmetrization. *IEEE Transactions on Wireless Communications* **13**(3), 1388–1396 (2014)
9. Liu, L., Ma, X., Ye, F., Ren, J.: Design of highly-parallel, 2.2Gbps throughput signal detector for MIMO systems. *IEEE International Conference on Communications* pp. 742–745 (2008)
10. Lofgren, J., Nilsson, P.: On MIMO K-Best Sphere Detector architecture complexity reductions. *2nd International Conference on Signal Processing and Communication Systems, ICSPCS 2008 - Proceedings* pp. 1–9 (2008)
11. Marzetta, T.L.: Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Transactions on Wireless Communications* **9**(11), 3590–3600 (2010)
12. Ngo, H.Q., Larsson, E.G., Marzetta, T.L.: Energy and spectral efficiency of very large multiuser MIMO systems. *IEEE Transactions on Communications* **61**(4), 1436–1449 (2013)
13. Nguyen, N.T., Lee, K.: Groupwise neighbor examination for tabu search detection in large MIMO systems. *IEEE Transactions on Vehicular Technology* **69**(1), 1136–1140 (2020)
14. Nguyen, N.T., Lee, K., Dai, H.: QR-Decomposition-aided tabu search detection for large MIMO Systems. *IEEE Transactions on Vehicular Technology* **68**(5), 4857–4870 (2019)
15. Sah, A.K., Chaturvedi, A.K.: Sequential and Global Likelihood Ascent Search-Based Detection in Large MIMO Systems. *IEEE Transactions on Communications* **66**(2), 713–725 (2018)
16. Srinidhi, N., Datta, T., Chockalingam, A., Rajan, B.S.: Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance. *IEEE Transactions on Communications* **59**(11), 2955–2963 (2011)
17. Srinidhi, N., Mohammed, S.K., Chockalingam, A., Rajan, B.S.: Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search. *IEEE International Symposium on Information Theory - Proceedings* pp. 1993–1997 (2009)
18. Srinidhi, N., Mohammed, S.K., Chockalingam, A., Rajan, B.S.: Near-ML Signal Detection in Large-Dimension Linear Vector Channels Using Reactive Tabu Search (2009). DOI <http://arxiv.org/abs/0911.4640>
19. Vardhan, K.V., Mohammed, S.K., Chockalingam, A., Rajan, B.S.: A low-complexity detector for large MIMO systems and multicarrier CDMA systems. *IEEE Journal on Selected Areas in Communications* **26**(3), 473–485 (2008)
20. Zhao, H., Long, H., Wang, W.: Tabu search detection for MIMO systems. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC* pp. 1–5 (2007)