A Comprehensive Comparison Between a Novel Tree-Search Algorithm and the Sphere-Decoding Based Algorithms

Chong Xu and Hamid Gharavi

National Institute of Standards and Technology, Gaithersburg, USA

Email: {chong.xu, hamid.gharavi}@nist.gov

Abstract

In this report we firstly propose a block-code based general model to combat the Inter-Symbol Interference (ISI) caused by frequency selective channels in a Multi-Input Multi-Output (MIMO) system and/or by asynchronous cooperative transmissions. The general model is not only exemplified by the Time-Reversed Space-Time Block Code (TR-STBC) scheme, but also by the Asynchronous Cooperative Liner Dispersion Codes (ACLDC) scheme. In these schemes a guard interval has to be inserted between adjacent transmission blocks to mitigate the effect of ISI. Consequently, this could degrade the effective symbol rate for a small block size. A larger block size would enhance the effective symbol rate and also substantially increase the decoding complexity. In the general model proposed in this paper, we further present a novel low-complexity breadth-adjustable tree-search algorithm and compare it with Sphere-Decoding (SD) based algorithms. With simulation results we will illustrate that our algorithm is able to achieve the optimal performance in terms of Bit Error Rate (BER) with a complexity much lower than the SD-based algorithms, whether the ACLDC or TR-STBC scheme is employed. Through simulation we further demonstrate that when the block size of the ACLDC is equivalent to 20, the complexity of the proposed algorithm is only a fraction of 10^{-8} that of the Maximum Likelihood (ML) algorithm. This would allow us to practically enhance the effective symbol rate without any performance degradation.

I. Introduction

Inter-Symbol Interference (ISI) can be caused by many different reasons. In this paper we mainly deal with the ISI incurred by frequency selective fading in Multi-Input Multi-Output (MIMO) systems and/or by asynchronous transmissions in cooperative networks. To handle either case, a number of block-based transmitting and receiving schemes, which contain T codewords per block, have been invented based on the structure of a single codeword of the space-time diversity scheme provided for channels without ISI. These block-based transceiver schemes aim at minimizing the effect of ISI while maintaining the same order of diversity, as achieved by their single-codeword counterparts in systems without ISI. Amongst them are Time-Reversed STBC (TR-STBC) [1] which is designed for MIMO systems with frequency selective channels, the shift group-decodable STBC [2] when synchronization cannot be achieved in a wireless relay network, as well as Linear Asynchronous STBCs (LA-STBCs) [3], [4] and Asynchronous Cooperative Linear Dispersion Codes (ACLDCs) [5] where both are devised to mitigate the ISI aroused by asynchronous cooperative transmissions. ACLDCs are more robust than LA-STBCs when the asynchronous delay τ is not an integral multiple of the symbol durations T_{s} [5]. Nevertheless, all the above mentioned techniques are only capable of removing the ISI between symbols within identical codewords. The ISI between symbols belonging to adjacent codewords has to be removed with the aid of non-linear detector/decoders. The optimal detector is the Maximum Likelihood (ML) detector, which has been applied in [5]. As a guard interval with a fixed length has to be inserted between any two contiguous blocks, a larger block size will result in a higher effective transmission rate. However when a larger block size is adopted, the complexity imposed by the ML detector becomes unaffordable [5]. The authors of [4]–[6] have mentioned that lattice-decoding algorithms can be used to decode their respective schemes. However, no details of the algorithms were provided.

In order to combine the above-mentioned various state-of-the-art transceiver schemes developed to combat ISI, we firstly propose a general model covering from the un-coded information symbols to the objective function of the detection algorithms. Then our main contribution is to design a novel low complexity tree-search detection algorithm based on the proposed general model.

Tree-search algorithms have been extensively applied to data compression and error correction [7]. One major category of reduced-state tree-search algorithms is the QRD-M algorithm [8]–[10] family. In this

case, premature pruning of branches may result in an error propagation, which needs to be mitigated using other procedures [11] and this could consequently add to computation complexity. Another significant branch of techniques is the Sphere Decoder (SD)-based algorithms [12]–[16] that were considered for frequency-selective MIMO channels. However, the complexity of the SD-based algorithms may become exponential with the block length under the worst case scenario [12]. Against this background, the authors of [13] have proposed two improved SD-based algorithms to combat the high complexity issue that could occur under the worst case scenario.

The goal of all the SD-based Algorithms (SDAs), such as [12], [13], is to constrain the search to only those candidates that lie inside a hypersphere with a radius r. Bearing in mind that Objective Function (OF) is defined to evaluate the merit of a trial solution, the OFs of SDAs have to provide the optimal solution with minimum OF output (fitness value) among the candidate set. The radius r is always set as the overall fitness value of the temporal optimal solution found so far. When the accumulated fitness value of a partial path calculated to a certain node exceeds r, the subtree originating from it can be pruned to save complexity. No path continuing from that node can be superior to the current best solution with an overall fitness value of r. This pruning process cannot be achieved if the optimal final solution has the maximum overall fitness value. Without any sub-tree pruned, SDAs are unable to save any complexity compared with the exhaustive searching algorithms.

In this paper, we propose a novel tree-search based algorithm that employs an OF providing the optimal solution with the maximum overall fitness value, which imposes a significantly lower complexity than the OF of the SDAs. Besides, our algorithm will not search from any sister branch at any level if the current partial path is only made up of states ranked first in each column. This can significantly reduce the overall computation complexity without loss of performance. Last but not least, in our approach the searching range M is user-defined and can be adjusted with the least computation complexity. All these features make our algorithm achieve the same Bit-Error Rate (BER) as obtained by the exhaustive searching algorithm, with a complexity not only lower than Viterbi Algorithm (VA) or ML, but also lower than the SDAs in [12], [13]. In Section II-A we firstly present the general system model that amalgamates ISI generated due to both reasons, which are asynchronous transmissions or/and frequency selective channels. Then, we specify the model for both STBC and ACLDC schemes. Details of our proposed algorithm, as

well as the SD-based algorithms, will be presented in Section III, where comparisons are also provided. We will present the simulation results in Section IV, followed by closing remarks in Section V.

II. System Model

A. General Form

In a system with ISI created either by the frequency selective channels or by the asynchronization of the cooperative nodes, we consider a block of T diversity-oriented codewords. In a MISO system with K transmit antennas, we will now quantify the unified transceiver schemes with a $(Q \times T)$-element input signal block $\mathbf{B} = [\mathbf{b}_1,\ \mathbf{b}_2,\ \ldots, \mathbf{b}_T]$. This is composed of T information vectors with each vector having Q information symbols, such that the block structure is able to combat the negative impact of the ISI and maintain the original diversity gain. Regardless of the transceiver schemes, each of the T original information vectors $\mathbf{b}_t^T = [b_{t1}, b_{t2},\ \ldots, b_{tQ}]$ will firstly be encoded to achieve the $(K \times \bar{T})$-element block $\mathbf{S}_t$, which can be expressed as:

$$\mathbf{S}_t = \Lambda(\mathbf{b}_t) = \left[\begin{array}{ccc} \mathbf{s}_1^{(t)} & \ldots & \mathbf{s}_{\bar{T}}^{(t)} \end{array}\right], \quad \forall t = 1, \ldots, T, \tag{1}$$

where the function $\Lambda(\cdot)$ characterizes the coding scheme and $\mathbf{s}_{\bar{t}}^{(t)} = \left[s_{1\bar{t}}^{(t)},\ s_{2\bar{t}}^{(t)},\ \ldots,\ s_{K\bar{t}}^{(t)}\right]^T$. When all the T blocks of $\mathbf{S}_t$ with $t = 1, \ldots, T$ are obtained, they will be transmitted in $\bar{T}$ successive time-blocks, each of which is composed of T consecutive symbol intervals. The T signal elements ranked in the same position of each encoded codeword, will be transmitted consecutively. More exactly, the $\bar{t}^{\text{th}}$ time-block will be used to transmit $s_{k\bar{t}}^{(1)},\ s_{k\bar{t}}^{(2)},\ \ldots,\ s_{k\bar{t}}^{(T)}$ at the kth antenna. When all the K number of antennas are considered, the $(K \times T)$-element signal matrix $\bar{\mathbf{S}}_{\bar{t}}$ to be transmitted from the K antennas over the T continuous symbol intervals, which are contained by the $\bar{t}^{\text{th}}$ transmitting blocks, can be expressed as: $\bar{\mathbf{S}}_{\bar{t}} = \left[\begin{array}{cccc} \mathbf{s}_{\bar{t}}^{(1)} & \mathbf{s}_{\bar{t}}^{(2)} & \ldots & \mathbf{s}_{\bar{t}}^{(T)} \end{array}\right], \forall \bar{t} = 1, \ldots, \bar{T}$. $\bar{\mathbf{S}}_{\bar{t}}$ can be regarded as the output of an operation manipulated on $\mathbf{S}_1$, $\mathbf{S}_2$,...,$\mathbf{S}_T$ generated in (1), achieving:

$$\bar{\mathbf{S}}_{\bar{t}} = \Theta_{\bar{t}}\Big(\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_T\}\Big), \tag{2}$$

where $\Theta_{\bar{t}}(\cdot)$ denotes the $\bar{t}^{\text{th}}$ operation function. Thus, the $(T \times 1)$-element received signal vector $\mathbf{r}_{\bar{t}}$ can be universally quantified as:

$$\mathbf{r}_{\bar{t}}^T = \mathbf{h}^T \mathbf{A} \bar{\mathbf{S}}_{\bar{t}} + \sum_{i \in \mathbb{I}} \tilde{\mathbf{h}}_i^T \tilde{\mathbf{A}}_i \tilde{\mathbf{S}}_{\bar{t}i} + \mathbf{n}_{\bar{t}}^T, \quad \forall \bar{t} = 1, 2, \ldots, \bar{T}, \tag{3}$$

where $\mathbf{A}$ and $\tilde{\mathbf{A}}_i$ are diagonal matrices with their diagonal elements denoting the amplitudes of the signals spread on different channels, $\mathbb{I}$ is the set of subscripts indicating the ISI components and $\mathbf{n}_{\bar{t}}$ denotes a $(T \times 1)$-element Gaussian noise vector with each element $\in \mathcal{N}(0, \sigma^2)$. The $(K \times 1)$-element vector $\mathbf{h}$ entails the CIR coefficients related to the channels conveying the desired signals. In (3), $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{S}}_{\bar{t}i}$ denote the channel coefficients and transmitted signal matrix related to the ith ISI component contaminating the signals received during the $\bar{t}^{\text{th}}$ phase.

The tth signal vector $\mathbf{z}_t$, that is fed into the user-defined detector, is normally the output of a linear signal processor Ω_t operated on all the received signals, which can be quantified as:

$$\mathbf{z}_t = \Omega_t\Big(\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{\bar{T}}\}\Big) = \mathbf{X}\bar{\mathbf{b}}_t + \sum_{i \in \mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\mathbf{b}}_{ti} + \bar{\mathbf{n}}_t, \quad \forall t = 1, 2, \ldots, T. \tag{4}$$

$\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ respectively denote the auxiliary matrix associated with the desired signal vector $\bar{\mathbf{b}}_t$, and the ith ISI component $\tilde{\mathbf{b}}_{ti}$, with respect to the tth decision input vector $\mathbf{z}_t$. As observed from (4), $\mathbf{z}_t$ can be regarded as the tth vector-element of $\mathbf{z}$. The solution corresponding to the entire signal transmitted during each block of T code-words can be solved by any optimization algorithm possessing the following OF:

$$E\big(\dot{\mathbf{B}}, \Omega(\mathbf{r}), \mathbb{I}\big) = \left\| \mathbf{z} - \underline{\mathbf{X}}\underline{\bar{\dot{\mathbf{b}}}} - \sum_{i \in \mathbb{I}} \underline{\tilde{\mathbf{X}}}_i \underline{\tilde{\dot{\mathbf{b}}}}_i \right\|^2, \tag{5}$$

where the $(Q \times T)$-element matrix $\dot{\mathbf{B}}$ is a trial estimation of the transmitted signal block, which can be expanded as $\dot{\mathbf{B}} = [\dot{\mathbf{b}}(1),\ \dot{\mathbf{b}}(2),\ \cdots,\ \dot{\mathbf{b}}(T)]$. $\underline{\mathbf{X}}$ and $\underline{\tilde{\mathbf{X}}}_i$ respectively denote the overall auxiliary matrix of the trial estimation of the desired transmitted signal $\underline{\bar{\dot{\mathbf{b}}}}$ and of the ith ISI component $\underline{\tilde{\dot{\mathbf{b}}}}_i$, when the entire block is considered. More exactly, we have $\underline{\mathbf{X}} = \mathbf{X} \otimes \mathbf{I}_T$, $\underline{\tilde{\mathbf{X}}}_i = \tilde{\mathbf{X}}_i \otimes \mathbf{I}_T$, $\underline{\mathbf{z}} = [\mathbf{z}_1^T,\ \mathbf{z}_2^T,\ \ldots,\ \mathbf{z}_T^T]^T$, $\underline{\bar{\dot{\mathbf{b}}}} = [\bar{\dot{\mathbf{b}}}^T(1),\ \bar{\dot{\mathbf{b}}}^T(2),\ \ldots,\ \bar{\dot{\mathbf{b}}}^T(T)]^T$, $\underline{\tilde{\dot{\mathbf{b}}}}_i = [\tilde{\dot{\mathbf{b}}}_i^T(1),\ \tilde{\dot{\mathbf{b}}}_i^T(2),\ \ldots,\ \tilde{\dot{\mathbf{b}}}_i^T(T)]^T$, where $\otimes$ represents the operation resulting in the Kronecker product. As can be observed, $\underline{\mathbf{X}}$ and $\underline{\tilde{\mathbf{X}}}_i$ are both second-level diagonal matrices with their T diagonal elements respectively being $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$. More exactly, corresponding to the different interpretation of the elements in $\mathbb{I}$, in a TR-STBC assisted system both $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ are $(\bar{T} \times Q)$-element diagonal matrices; while in an ACLDC assisted system, $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ are $(\bar{T} \times QK)$ and $(\bar{T} \times 2Q)$-element matrices respectively. A QR-decomposition is intended to transfer a $(N \times T)$-element rectangular auxiliary matrix into a $(T \times T)$-element triangular matrix, so that the overall OF can be represented as the sum of T local OFs and the tth last local OF has an input consisting of the trial estimations of the last t transmitted signals. Thus, the estimation of the entire $(T \times 1)$-element signal vector can be pursued on an

element-by-element basis with the aid of a T-column trellis table or a T-level tree. As just discussed, the auxiliary matrices in (5) are naturally diagonal under both system scenarios, hence a QR-decomposition is unnecessary in this case. The overall OF in (5) can be readily represented as the sum of T local OFs, which respectively calculate the local fitness values of the states in the T columns of the trellis table. The input of the tth local OF $\underline{\dot{\mathbf{b}}}(t)$ comprises the trial estimations of the N_{b} codewords centered around $\mathbf{b}_t$. Therefore, the T-codeword transmitted signal block can be estimated on a codeword-by-codeword basis, by employing a tree-search algorithm on a T-column trellis table or a T-level tree without manipulating the QR-decomposition. More explicitly, (5) can be represented as the superposition of T local OFs

$$E(\dot{\mathbf{B}}) = \sum_{t=1}^{T} e_t\big(\underline{\dot{\mathbf{b}}}(t), \mathbf{z}_t, \mathbf{X}, \tilde{\mathbf{X}}_i, \mathbb{I}\big) = \sum_{t=1}^{T} e_t\big(\bar{\dot{\mathbf{b}}}(t), \tilde{\dot{\mathbf{b}}}_i(t), \mathbf{z}_t, \mathbf{X}, \tilde{\mathbf{X}}_i, \mathbb{I}\big), \tag{6}$$

where $\bar{\dot{\mathbf{b}}}(t)$ and $\tilde{\dot{\mathbf{b}}}_i(t)$ are made up of elements selected from $\underline{\dot{\mathbf{b}}}(t)$. Further details related to the QR-decomposition will be addressed in Appendix A. Notations of formulae from (1) to (6) will be exemplified in the TR-STBC assisted system with frequency selective fading and the ACLDC assisted asynchronous cooperative system in Sections II-B and II-C below.

B. Time-Reversed Space Time Block Code

Now we will specify all the symbols throughout (1) to (6), when the time-reversed Space Time Block Code (TR-STBC) scheme [1] is adopted in the system with two transmit antennas and one receive antenna, where we have $\bar{T} = 2$, $Q = 2$ and $K = 2$. According to the STBC matrix $\mathcal{G}_2$, one symbol block is divided into two symbol intervals and the symbol $\mathbf{S}_t$ in (1) can be specified as:

$$\mathbf{S}_t = [\mathbf{s}_1^{(t)} \;\; \mathbf{s}_2^{(t)}] = \begin{bmatrix} b_{t1} & -b_{t2}^* \\ b_{t2} & b_{t1}^* \end{bmatrix}. \tag{7}$$

That is, when the TR-STBC scheme is adopted, a block of symbols $\mathbf{S}_t$, $\forall t = 1, \ldots, T$ can be divided into two blocks, $\mathbf{s}_1^{(t)}$, $\forall t = 1, \ldots, T$ and $\mathbf{s}_2^{(t)}$, $\forall t = 1, \ldots, T$. The transmission frame will also be divided into two halves. During the first half of the frame, b_{t1} will be transmitted from antenna one and b_{t2} will be transmitted from antenna two. During the second half of the frame, $-b_{t2}^*$ and b_{t1}^* will be transmitted in a time-reversed order. More explicitly, the $(2 \times T)$ vector $\bar{\mathbf{S}}_{\bar{t}}$ in (3) can be epitomised as:

$$\bar{\mathbf{S}}_1 = [\mathbf{s}_1^{(1)} \;\; \mathbf{s}_1^{(2)} \;\; \cdots \;\; \mathbf{s}_1^{(T)}] \qquad \bar{\mathbf{S}}_2 = [\mathbf{s}_2^{(T)} \;\; \mathbf{s}_2^{(T-1)} \;\; \cdots \;\; \mathbf{s}_2^{(1)}]. \tag{8}$$

Then, the $(2 \times T)$ vector $\tilde{\mathbf{S}}_{\bar{t}i}$ in (3) that precurs the ith ISI component to the tth received signal can be epitomised as:

$$\tilde{\mathbf{S}}_{1i} = [\mathbf{0}_{2\times i} \;\; \mathbf{S}_1^{(1)} \;\; \cdots \;\; \mathbf{S}_1^{(T-i)}] \quad \tilde{\mathbf{S}}_{2i} = [\mathbf{0}_{2\times i} \;\; \mathbf{S}_2^{(T)} \;\; \cdots \;\; \mathbf{S}_2^{(i+1)}], \tag{9}$$

where $\mathbf{0}_{2\times i}$ is a $(2 \times i)$-dimensional all-zero matrix. The $(L+1)$-tap frequency selective channel will be represented as a polynominal having an order of L. The simplest discrete-time model of an L-delay-tap frequency selective channel with two transmit antennas and one receive antenna can be quantified as:

$$\begin{aligned} r_t =& \frac{1}{\sqrt{L+1}} h_1(q^{-1}) b_{t1} + \frac{1}{\sqrt{L+1}} h_2(q^{-1}) b_{t2} + n_t \\ =& \frac{1}{\sqrt{L+1}} h_{10} b_{t1} + \frac{1}{\sqrt{L+1}} h_{11} b_{(t-1)1} + \cdots + \frac{1}{\sqrt{L+1}} h_{1L} b_{(t-L)1} \\ &+ \frac{1}{\sqrt{L+1}} h_{20} b_{t2} + \frac{1}{\sqrt{L+1}} h_{21} b_{(t-1)2} + \cdots + \frac{1}{\sqrt{L+1}} h_{2L} b_{(t-L)2} + n_t, \end{aligned} \tag{10}$$

where the power of the transmitted signal is assumed to be uniformly distributed among the $(L+1)$ frequency-selective sub-channels. The (2×1) vector $\mathbf{h}$ in (3) contains the first taps of the two independent frequency-selective channels, which can be further detailed as $\mathbf{h} = [h_{10}, h_{20}]^T$. Correspondingly, the ith tap of the polynomial model characterizing the channel with delay spread, as represented by $\tilde{\mathbf{h}}_i$ in (3), can be further detailed as $\tilde{\mathbf{h}}_i = [h_{1i}, h_{2i}]^T$. The two amplitude matrices in (3) can be epitomized as $\mathbf{A} = 1/\sqrt{2(L+1)}\mathbf{I}_2$ and $\tilde{\mathbf{A}}_i = 1/\sqrt{2(L+1)}\mathbf{I}_2$ respectively. The $(T \times 1)$ AWGN vector in (3) can be finalized as $\mathbf{n}_{\bar{t}} = [n_{1\bar{t}} \;, n_{2\bar{t}} \;, \cdots \; n_{T\bar{t}}]^T$.

On the receive side, the function Ω_t in (4) can be decomposed into two parts in the TR-STBC system. Firstly, the signal vector $\mathbf{r}_2$ containing the T samples collected during the second half frame have to be complex conjugated and time reversed in order to form the $(T \times 1)$-vector $\bar{\mathbf{r}}_2 = [\bar{r}_{12}, \; \cdots \; , \; \bar{r}_{T2}]$, where $\bar{r}_{t2} = r^*_{(T+1-t)2}$. Thus, the resultant matrix of the first part can be represented by $\bar{\mathbf{r}} = [\mathbf{r}_1^T \;\; \bar{\mathbf{r}}_2^T]^T$. Then, the

output matrix $\bar{\mathbf{r}}$ will be filtered with the matched filter $\mathbf{H}^H$ to generate the detection input:

$$\mathbf{z} = \begin{bmatrix} z_{11} & \cdots & z_{t1} & \cdots & z_{T1} \\ z_{12} & \cdots & z_{t2} & \cdots & z_{T2} \end{bmatrix} = \frac{1}{\sqrt{2(L+1)}}\mathbf{H}^H\bar{\mathbf{r}} = \frac{1}{\sqrt{2(L+1)}} \begin{bmatrix} h_1^*(q) & h_2(q^{-1}) \\ h_2^*(q) & -h_1(q^{-1}) \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \bar{\mathbf{r}}_2 \end{bmatrix}$$

$$\mathbf{z}_t = \begin{bmatrix} \frac{1}{\sqrt{2(L+1)}} \sum_{i=0}^{L} h_{1i}^* r_{(t+i)1} + \frac{1}{\sqrt{2(L+1)}} \sum_{i=0}^{L} h_{2i} \bar{r}_{(t-i)2} \\ \frac{1}{\sqrt{2(L+1)}} \sum_{i=0}^{L} h_{2i}^* r_{(t+i)1} - \frac{1}{\sqrt{2(L+1)}} \sum_{i=0}^{L} h_{1i} \bar{r}_{(t-i)2} \end{bmatrix} \tag{11}$$

Further deriving the above formula, we can obtain the detection input $\mathbf{z}_t$ in the form of (4), where each symbol will have its new definition specified in the TR-STBC scheme. More exactly, as for the desired signal, we have $\bar{\mathbf{b}}_t = [b_{t1},\, b_{t2}]^T$ and $\mathbf{X} = \frac{1}{2(L+1)} \sum_{k=1}^{2} \sum_{i=0}^{L} |h_{ki}|^2 \mathbf{I}_2$, As for the ISI components in (4), we will have $\tilde{\mathbf{b}}_{ti} = [b_{(t-i)1},\, b_{(t-i)2}]^T$, and the dynamic range for i can be divided into two closed integer areas which are $[-L, -1]$ and $[1, L]$ respectively. Correspondingly we will have $\tilde{\mathbf{X}}_i = \frac{1}{2(L+1)} \sum_{i=1}^{L} \sum_{k=1}^{2} \sum_{j=0}^{L-i} h_{kj}^* h_{k(j+i)} \mathbf{I}_2$, when $i = 1, \ldots, L$; and $\tilde{\mathbf{X}}_i = \frac{1}{2(L+1)} \sum_{i=-L}^{-1} \sum_{k=1}^{2} \sum_{j=0}^{L+i} h_{k(j-i)}^* h_{kj} \mathbf{I}_2$, when $i = -L, \ldots, -1$. The AWGN components in (4) can be finalized as $\bar{\mathbf{n}}_t = [\bar{n}_{t1}, \bar{n}_{t2}]^T$, where $\bar{n}_{t1} = 1/\sqrt{2(L+1)}\Big(\sum_{i=0}^{L} h_{1i}^* n_{(t+i)1} + \sum_{i=0}^{L} h_{2i} n_{(T+1-t+i)2}^*\Big)$ and $\bar{n}_{t2} = 1/\sqrt{2(L+1)}\Big(\sum_{i=0}^{L} h_{2i}^* n_{(t+i)1} - \sum_{i=0}^{L} h_{1i} n_{(T+1-t+i)2}^*\Big)$. As can be seen from the above analysis, in the TR-STBC assisted system the trial input of the local OF (6) can be expanded as: $\underline{\dot{\mathbf{b}}}(t) = [\dot{\mathbf{b}}(t-L),\, \ldots,\, \dot{\mathbf{b}}(t+L)]$, $\forall t = 1, \ldots, T$, where $\dot{\mathbf{b}}(-L+1) \equiv \cdots \equiv \dot{\mathbf{b}}(-1) \equiv \dot{\mathbf{b}}(T+1) \equiv \cdots \equiv \dot{\mathbf{b}}(T+L) \equiv \mathbf{0}_2$ are guard intervals.

C. Asynchronous Cooperative Linear Dispersion Code

Now we further exemplify our general system model previously addressed in Section II-A in an asynchronous cooperative system with K relay nodes. Among all the Linear dispersion code (LDC)-based system transceiver techniques, the ACLDC scheme proposed in [5] is able to combat the ill effect of ISI incurred by the asynchronous transmission. At the kth antenna, a transmit-antenna-specified $(\bar{T} \times Q)$-element coding matrix $\mathbf{C}_k$ is exploited to convert the original Q-element information vector $\mathbf{b}_t$ to $\bar{T}$ symbols. The rule of generating the coding matrix $\mathbf{C}_k$ is explained in [17] and some examples of $\mathbf{C}_k$ are given in the appendix of [5].

In spite of the values of parameters K, Q and $\bar{T}$, the coding matrix $\mathbf{C}_k$ can be uniformly expanded as: $\mathbf{C}_k = [\mathbf{c}_1^{(k)},\, \mathbf{c}_2^{(k)},\, \ldots,\, \mathbf{c}_{\bar{T}}^{(k)}]^T$ with the $(Q \times 1)$-element vector $\mathbf{c}_{\bar{t}}^{(k)}$ defined as $\mathbf{c}_{\bar{t}}^{(k)} = [c_{\bar{t}1}^{(k)},\, c_{\bar{t}2}^{(k)},\, \ldots,\, c_{\bar{t}Q}^{(k)}]^T$.

Thus the coding scheme characterized by function $\Lambda(\cdot)$ in (1) can be specified as: $\mathbf{S}_t = \Lambda(\mathbf{b}_t) = \bar{\mathbf{C}}\mathbf{B}_t$, where the symbols $\mathbf{S}_t$ and $\bar{\mathbf{C}}$ are $(K \times \bar{T})$- and $(K \times \bar{T}Q)$-element matrices that may be expanded as:

$$\mathbf{S}_t = \begin{bmatrix} s_{11}^{(t)} & \cdots & s_{1\bar{T}}^{(t)} \\ \vdots & \ddots & \vdots \\ s_{K1}^{(t)} & \cdots & s_{K\bar{T}}^{(t)} \end{bmatrix} \bar{\mathbf{C}} = \begin{bmatrix} \mathbf{c}_1^{(1)T} & \cdots & \mathbf{c}_{\bar{T}}^{(1)T} \\ \vdots & \ddots & \vdots \\ \mathbf{c}_1^{(K)T} & \cdots & \mathbf{c}_{\bar{T}}^{(K)T} \end{bmatrix} \tag{12}$$

Additionally $\mathbf{B}_t = \text{diag}\,(\mathbf{b}_t, \ldots, \mathbf{b}_t)$ is a $(\bar{T}Q \times \bar{T})$-element matrix, with $\mathbf{b}_t$ repeated $\bar{T}$ times on the diagonal positions. Comparing $\mathbf{S}_t$ in (12) with $\bar{\mathbf{S}}_t$ in (2), we may immediately retrieve the function $\Theta_{\bar{t}}$ in (2) as: $\Theta_{\bar{t}}(\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_T\}) = [\mathbf{s}_{\bar{t}}^{(1)}, \mathbf{s}_{\bar{t}}^{(2)}, \ldots, \mathbf{s}_{\bar{t}}^{(T)}]$, where the vector $\mathbf{s}_{\bar{t}}^{(t)}$ denotes the $\bar{t}^{\text{th}}$ $(K \times 1)$-element column vector of $\mathbf{S}_t$ in (12).

When the asynchronous cooperative system is considered, the channel vector $\mathbf{h}$ in (3) can be expanded as $\mathbf{h} = [h_1, h_2, \ldots, h_K]^T$, while the amplitude matrix $\mathbf{A}$ can be quantified as: $\mathbf{A} = \text{diag}\left([1, \sqrt{P_2}, \cdots, \sqrt{P_K}]\right)$, where $\sqrt{P_k}$ quantifies the power of the desired signal transmitted from the kth antenna maintained during the current symbol interval, $\forall k = 2, \ldots, K$. The ISI set $\mathbb{I}$ in (3) can be defined as $\mathbb{I} = \{2, 3, \cdots, K\}$. Symbols regarding the ISI components in (3) may be defined as:$\tilde{\mathbf{A}}_i = \text{diag}\left(\sqrt{P_{i,1}}, \sqrt{P_{i,2}}\right)$, $\tilde{\mathbf{h}}_i = [h_i, h_i]$ and $\tilde{\mathbf{S}}_{\bar{t}i} = [\ s_{i\bar{t}}^{(0)}, \ldots, s_{i\bar{t}}^{(T-1)};\ s_{i\bar{t}}^{(2)}, \ldots, s_{i\bar{t}}^{(T+1)}]$. $P_{i,1}$ and $P_{i,2}$ quantifies the power leaked to the previous and the next symbol intervals. The values of $P_{i,1}$ and $P_{i,2}$ are determined by the delay τ_i of the ith relay node with regard to the first node. Different values of P_1 and P_2 at a various delay amounts τ can be found in [5].

We may use a $(\bar{T} \times T)$-element matrix $\mathbf{R}$ to collectively define all the received vectors, yielding $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{\bar{T}}]^T$. As given in the ACLDC system [5], the signal processing function Ω_t in (4) can be interpreted as *getting the tth column of* $\mathbf{R}$. The auxiliary matrices $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ in (4) can be exemplified as: $\mathbf{X} = [h_1\mathbf{I}_{\bar{T}}, \sqrt{P_2}h_2\mathbf{I}_{\bar{T}}, \ldots, \sqrt{P_K}h_K\mathbf{I}_{\bar{T}}] \times \text{diag}\,(\mathbf{C}_1, \ldots, \mathbf{C}_K)$ and $\tilde{\mathbf{X}}_i = [\ \sqrt{P_{i,1}}h_i\mathbf{I}_{\bar{T}}, \sqrt{P_{i,2}}h_i\mathbf{I}_{\bar{T}}] \times \text{diag}\,(\mathbf{C}_i, \mathbf{C}_i)$, where $\mathbf{I}_{\bar{T}}$ denotes an $(\bar{T} \times \bar{T})$-element identity matrix. Additionally, the $(KQ \times 1)$-element desired signal vector $\bar{\mathbf{b}}_t$ and the ith $(2Q \times 1)$-element ISI signal vector $\tilde{\mathbf{b}}_{ti}$ may be denoted as: $\bar{\mathbf{b}}_t = [\mathbf{b}_t^T, \mathbf{b}_t^T, \ldots \mathbf{b}_t^T]^T$ and $\tilde{\mathbf{b}}_{ti} = [\mathbf{b}_{t-1}^T, \mathbf{b}_{t+1}^T]^T$. To constitute $\bar{\mathbf{b}}_t$, the identical $(Q \times 1)$-element vector $\mathbf{b}_t$ is repeated K times to match the definition of $\mathbf{X}$. Furthermore, to make up $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_T$, we will set $\mathbf{b}_0$ and $\mathbf{b}_{T+1}$ to $(Q \times 1)$-element all zero vectors, which is also known as the guard intervals and prevents ISI from

spreading to the adjacent blocks. As can be seen from the above analysis, in the ACLDC assisted system the trial input of the local OF (6) can be expanded as: $\underline{\dot{\mathbf{b}}}(t) = [\dot{\mathbf{b}}(t-1),\ \dot{\mathbf{b}}(t),\ \dot{\mathbf{b}}(t+1)]$, $\forall t = 1, \ldots, T$, where $\dot{\mathbf{b}}(0) \equiv \dot{\mathbf{b}}(T+1) \equiv \mathbf{0}_2$ are guard intervals.

III. Breadth Adjustable Tree-Search Algorithm

In this section, our decoding algorithm, which is termed as the 'breadth adjustable tree-search algorithm' (BATSA), will be presented and compared with the classical sphere-decoding-based algorithms (SDAs) presented in [12], [13], under both the ACLDC and the TR-STBC schemes assisted MISO system scenarios. The structure of the trees (or the trellis table) is merely decided by the transmission scheme and the system scenarios, but is independent of which decoding algorithm is employed. Our decoding algorithms (BATSA) as well as the SD-based algorithms will be elaborated in both the ACLDC and the TR-STBC assisted systems in terms of the following three aspects: *structure of the tree (trellis table)*, *objective functions* and *decoding procedures*, which will be detailed in the following text.

A. Structure of the Tree or Trellis Table

As mentioned before, any candidate solution can be represented by a $(Q \times T)$- (for ACLDC) or a $(1 \times T)$- (for TR-STBC) element matrix $\dot{\mathbf{B}}$. Nevertheless, in this section, we will unify the representation of the candidate solution with a $(1 \times T)$-element vector $\dot{\mathbf{v}}$. There will be N_{c} number of possible candidate solutions in the full set $\mathbb{V}$, where $N_{\text{c}} = 2^{QT}$ (for ACLDC) and $N_{\text{c}} = 2^{T}$ (for TR-STBC). We may index the candidate solution as $\dot{\mathbf{v}}_i$, with $i = 1, 2, \ldots, N_{\text{c}}$ by its vector value. Moreover, it can be expanded as: $\dot{\mathbf{v}}_i = [\dot{v}_{i,1},\ \dot{v}_{i,2},\ \ldots,\ \dot{v}_{i,T}]$.

Consider a T-level tree or a T-column trellis table. The node located at the jth row of the tth column can be denoted as (j, t) and the input trial vector of it can be represented by a N_{b}-element vector: $\mathbf{u}_j = [u_{j,1},\ u_{j,2},\ \ldots,\ u_{j,N_{\text{b}}}]$, while $N_{\text{b}} = 3$ (for ACLDC) or $N_{\text{b}} = 2L + 1$ (for TR-STBC). The implication of $\mathbf{u}_{j_t}$ can be interpreted by the elements of $\dot{\mathbf{v}}_i$ as

$$\begin{aligned} \mathbf{u}_{j_t} =& [\dot{v}_{i,t-1}, \dot{v}_{i,t}, \dot{v}_{i,t+1}] && \text{for ACLDC} \\ \mathbf{u}_{j_t} =& [\dot{v}_{i,t-L}, \dot{v}_{i,t-L+1}, \ldots, \dot{v}_{i,t}, \dot{v}_{i,t+1}, \ldots, \dot{v}_{i,t+L}] && \text{for TR-STBC,} \end{aligned} \tag{13}$$

where j_t is a real number ranging between $[1, N_{\rm r}]$ and $N_{\rm r} = 2^{3Q}$ (for ACLDC) or $N_{\rm r} = 2^{2L+1}$ (for TR-STBC). The value of j_t is decided by the value of vector $\mathbf{u}_{j_t}$. A valid path between $\mathbf{u}_{j_t}$ and $\mathbf{u}_{j_{t+1}}$ can be established, if $[u_{j_t,2}, \cdots, u_{j_t,N_{\rm b}}] = [u_{j_{t+1},1}, \cdots, u_{j_{t+1},N_{\rm b}-1}]$.

B. Objective Functions of BATSA and SDAs

1) Objective Function employed by SD-based algorithms: The OF employed by SD-based algorithms is the Euclidean distances that can be expanded from (6). As can be observed from (4), in the ACLDC system, the local OF of node $\mathbf{u}_j$ employed by all the SD-based algorithms can be quantified as:

$$e_t(\mathbf{u}_j, \mathbf{z}_t) = \left\| \mathbf{z}_t - \mathbf{X}\bar{\dot{\mathbf{b}}}_j - \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j \right\|^2 = \mathbf{z}_t^T \mathbf{z}_t - f_t(\mathbf{u}_j, \mathbf{z}_t), \tag{14}$$

where $f_t(\mathbf{u}_j, \mathbf{z}_t)$ is the local OF employed by the BATSA algorithm, which will be defined in (16) soon. In (14), $\bar{\dot{\mathbf{b}}}_j = [\digamma^T(u_{j,2}), \cdots, \digamma^T(u_{j,2})]^T$ with $\digamma(u_{j,2})$ repeated K times, and $\digamma$ represents the transfer function from a real number ranged within $[1, 2^Q]$ to a $(Q \times 1)$-element BPSK modulated signal vector. In (14), $\tilde{\dot{\mathbf{b}}}_j = [\digamma^T(u_{j,1}), \digamma^T(u_{j,3})]^T$. The definition of $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ has been given in Section II-C.

As for the TR-STBC assisted system, as can be summarized from the analysis in Section II-B and thanks to the special feature of the time-reversed strategy, when $\bar{T} = 2$, the decoder input $\mathbf{z}_t$ can be decomposed into two mutually independent variables z_{t1} and z_{t2}, which respectively contain the components of b_{t1} and b_{t2}, with $t = 1, 2, \ldots, T$. As a result, the local OF of node $\mathbf{u}_j$ employed by all the SD-based algorithms can be quantified as:

$$e_{t\bar{t}}(\mathbf{u}_j, z_{t\bar{t}}) = \left\| z_{t\bar{t}} - x\bar{\dot{b}}_j - \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i) \right\|^2 = z_{t\bar{t}}^* z_{t\bar{t}} - f_t(\mathbf{u}_j, z_{t\bar{t}}), \quad \bar{t} = 1, 2 \tag{15}$$

where the dynamic range for i can be divided into two closed integer areas which are $[-L, -1]$ and $[1, L]$ respectively. Correspondingly we will have $\tilde{x}_i = \dfrac{1}{2(L+1)} \sum_{i=1}^{L} \sum_{k=1}^{2} \sum_{j=0}^{L-i} h_{kj}^* h_{k(j+i)}$, when $i = 1, \ldots, L$; and $\tilde{x}_i = \dfrac{1}{2(L+1)} \sum_{i=-L}^{-1} \sum_{k=1}^{2} \sum_{j=0}^{L+i} h_{k(j-i)}^* h_{kj}$, when $i = -L, \ldots, -1$. Besides, in (17) $\bar{\dot{b}}_j = \digamma(u_{j,L+1})$ and $\tilde{\dot{b}}_{j,i} = \digamma(u_{j,L+1-i})$, for all $i = -L, \ldots, -1$ and $i = 1, \ldots, L$, where $\digamma$ represents the transfer function from a real number ranged within $[1, 2]$ to a BPSK modulated signal. $z_{t\bar{t}}^*$ is the conjugate of $z_{t\bar{t}}$ and $f_t(\mathbf{u}_j, z_{t\bar{t}})$ is the local OF employed by the BATSA algorithm, which will be defined in (17). Then the optimal solution can be obtained by solving: $\hat{\mathbf{v}} = \arg_{\hat{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_{\rm c}} \min \sum_{t=1}^{T} e_t(\mathbf{u}_{j_t}, \mathbf{z}_t)$

in ACLDC system or $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_{\rm c}} \min \sum_{t=1}^{T} e_t(\mathbf{u}_{j_t}, z_{t\bar{t}})$ in ACLDC system where the relationship between $\dot{\mathbf{v}}_i$ and $\mathbf{u}_{j_t}$ has been defined in (13).

2) Objective Function employed by BATSA: Alternatively, (6) can be simplified by eliminating the common parts independent of $\bar{\dot{\mathbf{b}}}(t)$ and $\tilde{\dot{\mathbf{b}}}(t,i)$. More exactly, in the ACLDC system, the local objective function (OF) of node $\mathbf{u}_j$ in the $t^{\rm th}$ column of hte trelis table employed by our decoding algorithm BATSA can be quantified as:

$$f_t(\mathbf{u}_j, \mathbf{z}_t) = 2\Re\left\{\left[\mathbf{X}\bar{\dot{\mathbf{b}}}_j + \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j\right]^H \mathbf{z}_t\right\} - \left[\mathbf{X}\bar{\dot{\mathbf{b}}}_j + \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j\right]^H \left[\mathbf{X}\bar{\dot{\mathbf{b}}}_j + \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j\right]. \tag{16}$$

The local OF of node (j,t) employed by our BATSA-based detector in a TR-STBC assisted system can be quantified as:

$$f_{t\bar{t}}(\mathbf{u}_j, z_{t\bar{t}}) = 2\Re\left\{\left[x\bar{\dot{b}}_j + \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i)\right]^* z_t\right\} - \left[x\bar{\dot{b}}_j + \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i)\right]^* \left[x\bar{\dot{b}}_j + \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i)\right], \quad \bar{t} = 1,2 \tag{17}$$

Then the optimal solution can be obtained by solving: $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_{\rm c}} \max \sum_{t=1}^{T} f_t(\mathbf{u}_{j_t}, \mathbf{z}_t)$, in an ACLDC assisted system or $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_{\rm c}} \max \sum_{t=1}^{T} f_t(\mathbf{u}_{j_t}, z_{t\bar{t}})$ in a TR-STBC assisted system, where the relationship between $\dot{\mathbf{v}}_i$ and $\mathbf{u}_{j_t}$ has been defined in (13).

C. Decoding Procedure of BATSA

The entire decoding procedure carried out by BATSA can be divided into two stages. The first stage is termed as the 'pre-processing' stage, and the second stage is the genuine searching process. The first stage can be further divided into two steps. During the first step of the first stage, the fitness value, i.e. output of the local OF of each node in the trellis table will be calculated. During the second step of the first stage, the $N_{\rm r}(t)$ nodes within the $t^{\rm th}$ column will be sorted in descending order according to their fitness values of the local OF.

The flow-chart of the entire searching process is depicted in Fig. 2. During the second stage, the searching algorithm can be divided into three steps: *establishing a complete route forwards*, *calculating the accumulated fitness value backwards* and *comparing and picking up the local elite vector*. Only one route is considered for any of these steps.

The first step, namely *establishing a complete route forwards* or '*extension*' as denoted in blue in Fig. 2, starts from the $1^{\rm st}$ column in a depth-first manner. If a route has been established, the initial searching

breadth M will not be changed. Otherwise, M will be increased by Δ. The searching breadth can thus be adjusted in the fastest manner. This step may be re-activated at any of the 2nd to the $(T-1)^{\text{th}}$ columns of the trellis table during the following steps. In contrast to the first step, the above-mentioned second and third steps, which are respectively marked as block of '*calculate partial path metric*' and of '*compare & pick up elite partial path*' in Fig. 2, will be operated in turn at any search-depth from the $(T-1)^{\text{th}}$ column backwards to the 2nd columns of the fitness table.

For the sake of presentation convenience, in the following, the first number, which usually contains parameter Q, is associated with the ACLDC system; while the second number, which is in a pair of angle brackets, is referred to the TR-STBC aided system.

The full set $\mathbb{V}$ can be divided into 2^{2Q} $\langle 2^{2L} \rangle$ subsets in $(T-1)$ ways, according to the value of the $(t-1)^{\text{th}}$ and the tth elements $\langle$the $(t-L)^{\text{th}}$, $(t-L+1)^{\text{th}}$, $\cdots$, $(t+L-1)^{\text{th}}$ elements$\rangle$ of the candidate solution vectors $\dot{\mathbf{v}}$, when $t = 2, \ldots, T$. The γth subset is denoted as $\mathbb{V}_{\gamma}^{(t)}$, and $\mathbb{V} = \bigcup_{\gamma=1}^{N_{\text{g}}} \mathbb{V}_{\gamma}^{(t)}$, with $N_{\text{g}} = 2^{2Q}$ $\langle N_{\text{g}} = 2^{2L} \rangle$. For each subset $\mathbb{V}_{\gamma}^{(t)}$, it can be further divided into $2^{Q(t-2)}$ $\langle 2^{t-L} \rangle$ subsets, recorded as $\mathbb{V}_{r,j}^{(t)}$, with $j = 1, \ldots, 2^{Q(t-2)}$ $\langle j = 1, \ldots, 2^{t-L} \rangle$.

Similarly, all the nodes within the tth column of the trellis table can be divided into $N_{\text{g}} = 2^{2Q}$ $\langle 2^{2L} \rangle$ groups, according to the value of their first 2 $\langle 2L \rangle$ elements. Obviously, all the nodes within the same group in the tth column emanate from the same node in the $(t-1)^{\text{th}}$ column of the trellis table. Thus, we may alternatively denote $\mathbf{u}_{j_t}$ as $\mathbf{u}_{\gamma_t, i_t}$, where $\gamma_t \in [1, N_{\text{g}}]$ and $i_t \in [1, N_{\text{u}}]$, with $N_{\text{u}} = 2^{Q}$ $\langle 2 \rangle$. There will be a unique pair $\{\gamma_t, i_t\}$ corresponding to each $j = 1, \ldots, N_{\text{r}}(t)$.

For a given T-element candidate solution vector $\dot{\mathbf{v}}_i$, the accumulated fitness value $\bar{F}_t(\dot{\mathbf{v}}_i)$ of the last $(T-t+1)$ elements is defined as$\bar{F}_t(\dot{\mathbf{v}}_i) = \sum_{t=t}^{T} f_t\big(\mathbf{u}_{j_t}\big)$, where the implication of $\mathbf{u}_{j_t}$ has been specified in (13). The vector $\bar{\mathbf{v}}_{\gamma,i}^{(t)}$ is defined as a T-element vector with its first $(t-2)$ $\langle (t-L-1) \rangle$ elements being zero, the $(t-1)^{\text{th}}$ and the tth elements $\langle$the $(t-L)^{\text{th}}$, $(t-L+1)^{\text{th}}$, $\cdots$, $(t+L-1)^{\text{th}}$ elements$\rangle$ equalling to $u_{j_t,1}$ and $u_{j_t,2}$ $\langle u_{j_t,1}, u_{j_t,2}, \cdots, u_{j_t,2L} \rangle$ and with the combination of its last $(T-t)$ $\langle (T-t-L+1) \rangle$ elements indexed by i. We further exploit notation $\bar{\mathbb{V}}_{\gamma}^{(t)}$ as the set collecting all the vectors $\bar{\mathbf{v}}_{\gamma,i}^{(t)}$, $\forall i = 1, \ldots, 2^{Q(T-t)}$ $\langle \forall i = 1, \ldots, 2^{T-t-L+1} \rangle$. We further define $\hat{\bar{\mathbf{v}}}_{\gamma}^{(t)}$ as the vector having the highest accumulated fitness value among all the vectors in set $\bar{\mathbb{V}}_{\gamma}^{(t)}$. Moreover, the elite accumulated fitness value $\hat{q}_{\gamma}^{(t)}$ is defined as the accumulated fitness value from the $(t+1)^{\text{th}}$ to the Tth element associated with $\hat{\bar{\mathbf{v}}}_{\gamma}^{(t)}$.

Based on the above notations, our algorithm can be manipulated according to the following instructions:

1) Do not begin to calculate and pick up until a route from the 1st to the Tth column has been established;
2) If the current column is not the Tth, establish a valid route;
3) Suppose a valid path has been established from $\mathbf{u}_{\gamma_t,i_t}$ to $\mathbf{u}_{\gamma_{t+1},i_{t+1}}$. After gaining $\hat{\bar{\mathbf{v}}}^{(t+1)}_{\gamma_{t+1}}$ from the $(t+1)^{\text{th}}$ column, affix it with the first digit of the current node, i.e. $u^{(t)}_{\gamma_t,i_t,1}$, and copy the elements from the tth to the Tth position of $\hat{\bar{\mathbf{v}}}^{(t+1)}_{\gamma_{t+1}}$ to $\bar{\mathbf{v}}^{(t)}_{\gamma_t,i_t}$;
4) Suppose a valid path has been established from $\mathbf{u}_{\gamma_t,i_t}$ to $\mathbf{u}_{\gamma_{t+1},i_{t+1}}$. After gaining $\hat{q}^{(t+1)}_{\gamma_{t+1}}$ from the $(t+1)^{\text{th}}$ column, accumulate the value with the local fitness value of the current node $f_t(\mathbf{u}_{\gamma_t,i_t})$ and record the sum as $q^{(t)}_{\gamma_t,i_t}$;
5) Before returning to the $(t-1)^{\text{th}}$ column, we need to re-activate 'Step 1 - *establishing a complete route forward*' from all the sister nodes belonging to the same group γ_t in the tth column within the searching breadth.
6) Assuming there are N_{u} number of nodes belonging to the γ_t^{th} group within the searching breadth, compare $q^{(t)}_{\gamma_t,\bar{i}}$ through $\bar{i}=1,\ldots,N_{\text{u}}$. Return to the $(t-1)^{\text{th}}$ column with the vector $\bar{\mathbf{v}}^{(t)}_{\gamma_t,k}$ having the highest $q^{(t)}_{\gamma_t,k}$ as $\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_t}$ in combination with $q^{(t)}_{\gamma_t,k}$ as $\hat{q}^{(t)}_{\gamma_t}$.

Eventually, the final solution $\hat{\mathbf{v}}$ is equivalent to $\hat{\bar{\mathbf{v}}}^{(1)}_{\hat{\gamma}_1}$, where $\hat{\gamma}_1=\arg\max_{\gamma_1\in\{1,\ldots,N_{\text{g}}\}}\{\hat{\bar{q}}^{(1)}_{\gamma_1}\}$, with $N_{\text{g}}=2^{2Q}$ $\left\langle N_{\text{g}}=2^{2L}\right\rangle$. The above procedure is precisely expressed with pseudo codes in Alg. 1 (see page 26). An illustration of the whole process can be found in Fig. 1, with a system employing an ACLDC transmitting scheme with $K=2$, $Q=2$, $\bar{T}=2$ and a search breadth $M=8$. Note that the numbers in blue or red indicate the order of the corresponding branch being visited or backtracked. The example in Fig. 1 shows that the first successfully established route indicated by Steps 1,2,3,4 and 5 is '$032_1\rightarrow 321_2\rightarrow 211_3\rightarrow 114_4\rightarrow 142_5\rightarrow 420_6$'. Below we will use the form of '321_2' to represent Node $\mathbf{u}_{14,1}=[3\ 2\ 1]$ in the 2nd column of the trellis table and $032_1\rightarrow 321_2$ as the transit from State 032_1 to State 321_2. The fitness value $f_6([420])$ and $f_5([142])$ will be accumulated in backtracking Steps 6 and 7 to Level 4. Then, it will explore a new path spawning from '114_4' through Steps 8 and 9. After backtracking the partial path indicated by Steps 8, 9 to node '114_4' again, the algorithm starts searching for another possible valid path through Step 12. The resultant Node '144_5' has no valid branch to any node in Level 6 within the

searching breadth. Hence, Step 12 is marked as an 'unsuccessful extension' with a dashed line. So far, all the possible branches spawning from Node '114_4' have been exhausted within the current allowed search breadth. A comparison will be made among the accumulated fitness values of all the partial solutions that have been found so far: $\bar{F}_5([000142]) = f_5([142]) + f_6([420])$ and $\bar{F}_5([000143]) = f_5([143]) + f_6([430])$. Since $\bar{F}_5([000142]) > \bar{F}_5([000143])$, the partial route '$114_4 \rightarrow 142_5 \rightarrow 420_6$' wins out and is determined as the elite partial route, yielding $\hat{\bar{\mathbf{v}}}_4^{(5)} = [000142]$ and $\hat{q}_4^{(5)} = \bar{F}_5([000142])$. Correspondingly, the elite partial route '$114_4 \rightarrow 142_5 \rightarrow 420_6$' represented by Steps 4 and 5, are also highlighted by bold lines in Fig. 1.

This process continues until $q_2^{(1)}$ is calculated as $F(\hat{\bar{\mathbf{v}}}_{14}^{(2)}) = f_1([042]) + \hat{q}_{14}^{(2)}$ in Step 44. As can be observed from Figs. 1, there are no more nodes in Column 1 that can be used as a root originating a valid transit to any node in Column 2 within the searching breadth. As a result, all the searching processes allowed by the current searching breadth $M = 8$ have run out. As indicated by Lines 12 and 13 of Alg. 2 (see page 32), the final solution can be determined as the vector having the highest overall fitness value between $\bar{\mathbf{v}}_1^{(1)}$ and $\bar{\mathbf{v}}_2^{(1)}$. In the given example in Figs. 1, the final solution is decided as $\bar{\mathbf{v}}_2^{(1)} = [422432]$ as $q_2^{(1)} > q_1^{(1)}$.

D. Complexity Reduction of BATSA

Two kinds of operations are considered to further reduce the complexity of the tree search algorithm presented in Section III-C: *Straight top break block* and *breadth-adjustment scheme*. The former will reduce the complexity of the algorithm under a special case and will not degrade the bit-error-rate (BER) performance. The latter, as one of the major novelties of our algorithm, will reduce the complexity of the algorithm under all cases at the expense of an insignificant performance loss.

1) Straight-Top Break (STB): A non-performance-loss complexity-reduction procedure, called *Straight Top Break* (STB) block is designed to further reduce the complexity of the optimal or suboptimal tree-search algorithm. The main rationale to activate STB at the tth column of the fitness table, is that if so far the current partial route $\bar{\mathbf{v}}_{\gamma,i}^{(t)}$ is made up of nodes ranked at the first row in each column for $\dot{t} = t, \ldots, T$, there is no need to run the search function in the current t^{th} column from any other node. Ultimately, a significant amount of unnecessary complexity imposed by those potential transitions impossible to be

parts of the final solution can be saved.

2) Breadth Adjustment: When the channel is not extremely hostile, some complexity might be further saved by limiting the search and backtracking within the first M rows of the trellis fitness table. The depth-first search allows us to make a quick decision to increase the M by a step of Δ, if no valid route has been found within the current search-breadth M.

As can be observed from the example shown in Fig. 1, by setting the searching breadth $M = 8$, many unnecessary searches and calculations are averted, thereby easing the overall complexity. Nevertheless, it is also possible that some nodes constituting the optimal solution are out of the range designated by the searching breadth. Under such circumstances, if another sub-optimal solution has already been found within the available searching bound, the searching breadth will not be increased. As a result, the optimal solution will be omitted, incurring a performance loss. Nonetheless, we may find a tradeoff between the system BER performance and the computation complexity to maximize the efficiency of the decoding algorithm. More discussion about this issue can be found in Section IV.

E. Decoding Procedure of SDAs

In this subsection, we will briefly review the procedures of the SDAs proposed in [12], [13]. The search algorithm provided in [12] will be denoted as SD below, and the SDAs presented in [13] will be represented as VA/SD in the following text. Suppose the elite vector found so far can be denoted as $\hat{\mathbf{v}}_i$, and its corresponding route in the trellis table is $\{\hat{\mathbf{u}}_{j_1}, \hat{\mathbf{u}}_{j_2}, \ldots, \hat{\mathbf{u}}_{j_T}\}$. Then the constraint radius can be defined as $r = \sum_{t=1}^{T} e_t(\hat{\mathbf{u}}_{j_t})$, where the definition of e_t can be found in (14) (for ACLDC) and (15) (for TR-STBC). Unlike BATSA, the accumulated fitness value is calculated in the time sequential order from $t = 1, \ldots, T$ in the SDAs. During the continuing searching process, the following constraint has to be fulfilled for all $t \in [1, T]$:

$$\sum_{t=1}^{t} e_t(\mathbf{u}_{j_t}, \mathbf{z}_t) \leqslant r \tag{18}$$

If node $\mathbf{u}_{j_t}$ violates (18), the subtree emanating from it can be pruned. The searching radius r can be further reduced, if a superior vector having a smaller overall fitness value than $\hat{\mathbf{v}}_i$ has been found out.

Apart from the above features, the VA/SD proposed in [13] will also make use of the Markovian properties of the channel and will also prune the subtrees from node $\mathbf{u}_{\gamma_t,i}$. This occurs if the elite

accumulated fitness value, which is calculated from T to t, among all the partial paths represented by the sub-trees emanating from $\mathbf{u}_{\gamma_t,i}$ is bigger than that of any partial path emanating from $\mathbf{u}_{\gamma_t,j}$. $\mathbf{u}_{\gamma_t,j}$ is the sister node of $\mathbf{u}_{\gamma_t,i}$, and both of $\mathbf{u}_{\gamma_t,j}$ and $\mathbf{u}_{\gamma_t,i}$ emanate from the same node in the $(t-1)^{\text{th}}$ column.

F. Difference between BATSA and SDAs

The first difference between our proposed BATSA algorithm and the SDAs lies in the OF. Based on the principle described in Section III-E, the OF employed by all SD-based algorithms provide the optimal solution with the minimum OF value among all the candidate solutions, such as (14) (for ACLDC) and (15) (for TR-STBC). On the other hand, the OF employed by our BATSA algorithm such as (16) (for ACLDC) and (17) (for TR-STBC) would require the maximum OF value for the optimal solution, which consequently reduces the complexity in calculating the local OFs.

The computational complexity of the local OFs is the superposition of the real-number FLOPS imposed by calculating the local OF of all the $N_{\text{r}}T$ number of states in the trellis table during the detection procedure for a block of signal, given that there are N_{r} rows and T columns in the trellis table. It should be noted that $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ are decided entirely by channel impulse responses, regardless of the state during the calculation of the local OFs. Thus, we will not consider the complexity of calculating $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$, when we compare the difference of complexity imposed by local OFs employed by the BATSA and the SDAs. Apart from $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$, we have listed other inputs of the local OFs in the Table I.

TABLE I: Inputs of the local OF associated with each state in the $(N_{\text{r}} \times T)$-dimensional trellis table.

j \ t	1	2	$\cdots$	T
1	$\mathbf{u}_1, \mathbf{z}_1$	$\mathbf{u}_1, \mathbf{z}_2$	$\mathbf{u}_1, \cdots$	$\mathbf{u}_1, \mathbf{z}_T$
2	$\mathbf{u}_2, \mathbf{z}_1$	$\mathbf{u}_2, \mathbf{z}_2$	$\mathbf{u}_2, \cdots$	$\mathbf{u}_2, \mathbf{z}_T$
$\vdots$	$\vdots, \mathbf{z}_1$	$\vdots, \mathbf{z}_2$	$\vdots, \cdots$	$\vdots, \mathbf{z}_T$
N_{r}	$\mathbf{u}_{N_{\text{r}}}, \mathbf{z}_1$	$\mathbf{u}_{N_{\text{r}}}, \mathbf{z}_2$	$\mathbf{u}_{N_{\text{r}}}, \cdots$	$\mathbf{u}_{N_{\text{r}}}, \mathbf{z}_T$

For the sake of convenience, we may define $\mathbf{a} = \mathbf{X}\bar{\dot{\mathbf{b}}}_j + \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j$ (for ACLDC) and $a = x\bar{\dot{b}}_j + \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i)$ (for TR-STBC). Please note that $\mathbf{a}$ has to be calculated in both (14) and (16) in the current

version. Therefore, in this respect both (14) and (16) have the same computational complexity. Furthermore, as can be observed from (14) or (16), $\mathbf{a}$ does not depend on the value of $\mathbf{z}_t$ and it only has N_{r} number of legitimate values corresponding to the N_{r} legitimate values of $\mathbf{u}_j$, when $j = 1, \ldots, N_{\mathrm{r}}$ (see Table I). It should be noted that both $\bar{\dot{\mathbf{b}}}_j$ and $\tilde{\dot{\mathbf{b}}}_j$ are determined by $\mathbf{u}_j$ where their relationship is addressed after (14). Overall, the value of $\mathbf{a}$ and the second term of (16), which is $(\mathbf{a}^H\mathbf{a})$, only have N_{r} legitimate values. Hence, the calculation of $\mathbf{a}$ or $(\mathbf{a}^H\mathbf{a})$ need only to be operated N_{r} times $(j = 1, \ldots, N_{\mathrm{r}})$ throughout the entire trellis table.

It is important to note that all the FLOPS are calculated in terms of real-number operations. Therefore, one complex-multiplication equals four real-number multiplications and two real-number additions. Besides, if the additions are between two $(N \times 1)$-vectors, the number of FLOPS imposed by the vector-addition will be N times of that required by each element. Similarly, the number of FLOPS imposed by multiplication between two $(N \times 1)$-vectors will be N times of that required by each element, with an extra of $2(N-1)$ real-number additions.

First of all, let's look at the local OFs (16) and (14) employed by BATSA and SDAs in the ACLDC assisted system. Let's firstly consider the operations in (16). As suggested by the reviewers, (16) can be decomposed into two operations: $\Re\{\mathbf{a}^H\mathbf{z}_t\}$ and $\mathbf{a}^H\mathbf{a}$. When the value of the $(\bar{T} \times 1)$-element vector $\mathbf{a}$ is available, the calculation of $(\mathbf{a}^H\mathbf{a})$ will impose 2 real-number multiplications, 1 real-number addition per element[1], yielding $3\bar{T}$ FLOPS per index j. This further results in $3\bar{T}N_{\mathrm{r}}$ FLOPS per vector per trellis table required by (16) but not (14).

Although the first part of (16) - $\Re\{\mathbf{a}^H\mathbf{z}_t\}$ - is a vector-multiplication and its calculation has to be manipulated $N_{\mathrm{r}}T$ times, only the real part of the result will be needed. Each element multiplication will cost 3 real-number FLOPS[2]. Therefore, the FLOPS imposed by $\Re\{\mathbf{a}^H\mathbf{z}_t\}$ is $(3\bar{T} + \bar{T} - 1)$ per state, and $(3\bar{T} + \bar{T} - 1)N_{\mathrm{r}}T$ per trellis table. In summary, the complexity imposed by (16) apart from calculating $\mathbf{a}$, is $3\bar{T}N_{\mathrm{r}} + (3\bar{T} + \bar{T} - 1)N_{\mathrm{r}}T$ real-number FLOPS. All these complexity results are recorded in Table II.

On the other hand, in (14) the calculation imposed by operation $\mathbf{d} = \mathbf{z}_t - \mathbf{a}$ and $\|\mathbf{d}\|^2$ as stated by

[1] This is because each element of $\mathbf{a}$, if represented by $a + bj$, will take part in a calculation of $(a - bj)(a + bj) = a^2 + b^2$. So there are two real-number multiplications and one real-number additions.

[2] If there are two complex-valued numbers $a + bj$ and $c + dj$, $\Re\{(a + bj)(c + dj)\} = \Re\{ac - bd + (bc + ad)j\} = ac - bd$. Therefore calculating $\Re\{(a + bj)(c + dj)\}$ will impose three real-number FLOPS.

the reviewer have to be carried out $N_{\rm r}T$ times per trellis table. Firstly, the real-number FLOPS imposed by calculating $\mathbf{d} = \mathbf{z}_t - \mathbf{a}$ is $2\bar{T}$ real number additions per state. Secondly, calculating $\|\mathbf{d}\|^2$ will cost an extra of $(3\bar{T} + \bar{T} - 1)$[3] real-number FLOPS per state. **Therefore, in terms of the entire trellis table, the FLOPS imposed by (14) - apart from that by calculating $\mathbf{a}$ - is $(5\bar{T} + \bar{T} - 1)N_{\mathbf{r}}T$, which is $\delta = 2\bar{T}N_{\mathbf{r}}T - 3\bar{T}N_{\mathbf{r}}$ more than that by (16).** As also been elaborated in Table II.

As we can see, in ACLDC assisted system, as long as we have $\delta = 2\bar{T}N_{\rm r}T - 3\bar{T}N_{\rm r} > 0$, which results in $T > 3/2$, we will have the complexity of calculating (14) more than that of (16). As a matter of fact, having a block of more than one codeword is normally assumed by default, hence the condition $T > 3/2$ can be easily fulfilled. Therefore, we have shown that the FLOPS imposed by calculating the local OF of all states in the entire trellis table employed by the SD-based algorithms, is more than calculating the OF employed by our algorithm in the ACLDC system, as long as the block-length $T > 1$.

Similarly, as for the local OFs employed by our BATSA in the TR-STBC assisted system, apart from a, there will be $3N_{\rm r}T$ real-number FLOPS imposed by calculating $\Re\{a^* z_t\}$, and $3N_{\rm r}$ real-number FLOPS imposed by calculating a^*a in (17), in terms of the entire trellis table.

On the other hand, as for the local OF (15) employed by SDAs, there will be $2N_{\rm r}T$ real-number FLOPS imposed by calculating $\mathrm{d} = z_{t\bar{t}} - a$, and $2N_{\rm r}T$ real-number FLOPS imposed by $\|\mathrm{d}\|^2$, in terms of the entire trellis table. Apart from the FLOPS imposed by the above-mentioned operations, we may notice that either (15) or (17) quantifies the local OF based only on one of the $\bar{T}$ receive signals during the entire T-codeword block. Thanks to the structure of the TR-STBC scheme, detection can be manipulated in parallel with $\bar{T}$ independent detectors. However, the FLOPS of one detector has to be multiplied by $\bar{T}$ times in terms of the complete detection operation, though $\mathbf{a}$ has to be calculated only $N_{\rm r}$ times in the entire course. All of these have been illustrated in Table II.

Therefore, in terms of the complete detection procedure, the FLOPS imposed by calculating the OF of the SD-based algorithms in a TR-STBC assisted system is $(2N_{\mathbf{r}}T\bar{T} - 3N_{\mathbf{r}})$ more than that by our algorithms. As can be observed, as long as we have $N_{\rm r}(2T\bar{T} - 3) > 0$, which results in $T\bar{T} > 3/2$, we may draw to the conclusion that calculating (15) costs more complexity than calculating

[3]This is because each element of $\mathbf{d}$, if represented by $a + bj$, will take part in a calculation of $(a - bj)(a + bj) = a^2 + b^2$. So there are two real-number multiplications and one real-number additions.

(17). Obviously, the condition $T\bar{T} > 3/2$ is guaranteed to be satisfied in a TR-STBC assisted system having multiple transmit antennas. As a summary, the complexity of calculating (15) is always higher than that of (17) when a MIMO system employs TR-STBC schemes.

The other differences have been scattered in the previous text, and will be summarized in Table III. We should mention that in this table, the restriction of the searching breadth to a certain level will sometimes provide the system with a near-optimal BER at a significantly reduced complexity, which will be proved by simulation results in Section IV-C. Based on the above analysis, the complexity of our algorithm will be lower than any SD-based algorithms. This is further proved by the simulation results shown in Section IV.

G. Proof of Optimal Solution

We will demonstrate that our algorithm BATSA will provide the optimal solution by proving the following theorem:

***Theorem** 1:* When the algorithm jointly presented by Algs. 1 (see page 26) and 2 (see page 32) is employed with $M = 2^{3Q}$ (for ACLDC) or $M = 2^{2L+1}$ (for TR-STBC), the decoder output $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}} \Big\{ F(\dot{\mathbf{v}}) \Big\}$.

Proof: See Appendix B. ■

IV. Simulation Results

In this section, the BER and complexity performance of our algorithm will be denoted as 'BA' as a short for 'Breadth Adjustable'. It will be investigated in various system scenarios which employ diverse ACLDC schemes or TR-STBC schemes. The performances of our algorithm will be compared with the SD algorithm [12] and the VA/SD algorithm [13]. The effect of the block length T and the searching breadth M to our algorithm will also be studied.

A. Effect of the ACLDC Block-Size T

As can be seen in Fig. 3, our tree-search algorithm with $M = 64$ is shown to provide the system with the optimal solution, when ACLDC scheme with $K = 2$, $\bar{T} = 2$, $Q = 2$ (which is short for ACLDC(2,2,2))is employed. No difference can be observed between it and the SD algorithm or the VA/SD algorithm. It

also overlaps with the curve obtained by the ML algorithm, though the performance of the ML algorithm was unattainable when the block size exceeds $T = 8$ due to an unrealistic long simulation run time. When the block size increases from 2 to 20 at a step of 2, the effective symbol rate is 0.67, 0.8, 0.85, 0.88, 0.909, 0.923, 0.933, 0.941, 0.947, 0.952. Based on the above observation, we may conclude that, given a delay value τ, both the effective symbol rate and the BER performance are improved along with the increased block-size T. Additionally, the BER performance of the system is also advanced when a larger searching breadth is allowed. However, the superior strength of the algorithm with a larger M value over that with smaller searching breadth, is decreased as block-size T increases.

The simulated complexity results versus block-size T are illustrated in Fig. 4, which is calculated as the average number of FLOPS, including the number of additions, multiplications and comparisons, cost by each symbol to decode the entire ACLDC block. As can be observed in Fig. 4, while the complexity consumed by each symbol soars exponentially with block-size T when the ML algorithm is put to use. It severely limits the system from achieving a higher efficiency and better BER performance. On the contrary, the complexity imposed by BA remain almost the same in spite of the block-size increment. The difference between the expected FLOPS per symbol cost by algorithms with diverse initial searching breadth M and incremental steps Δ is almost invisible in this figure, compared with that between any of the BATSA and the ML algorithm. However, as can be observed from Fig. 4, the VA/SD algorithm saves more complexity than the SD-based algorithm, and our algorithm saves even more complexity than the VA/SD algorithm. The difference of the complexity between our algorithm and the VA/SD algorithm can be observed with higher resolution in Figs. 5 and 6, which respectively depict the Probability Density Function (PDF) of the FLOPS cost per block by a VA/SD algorithm and by our algorithm.

B. Investigation in Different Systems

In this subsection, we will investigate the BER and complexity performances of the VA/SD algorithm and our algorithm in different systems. More exactly, the BER performance versus the system SNR obtained in a flat-fading system employing an ACLDC$(2, 2, 2)$ scheme or an ACLDC$(2, 2, 3)$ scheme have been illustrated in Fig. 7. Besides, the BER versus SNR performance of a TR-STBC assisted system with three-path or four-path frequency selective channels, has also been depicted in Fig. 7. As can be

observed from Fig. 7, no matter what system is the environment, the BER performance of our algorithm does not show any difference from that achieved by the VA/SD algorithm. We will present the complexity cost of both algorithms in the TR-STBC system in Fig. 8. As can be observed from Fig. 8, our algorithm is always more economic than the VA/SD algorithm when the system is associated with various parameters.

C. Influence of the Searching Breadth M

As noted from Fig. 9, with a fixed incremental step of $\Delta = 2$, the BER of a system employing ACLDC(2,2,2) drops significantly with the enhancement of the initial searching breadth M when $M < 20$. However, no further improvement of BER has been observed when the initial searching breadth M exceeds 20. The curve quantifying the PDF of the FLOPS per block, achieved with searching breadth, gradually increases from $M = 4$ to $M = 64$ and is depicted as a three-dimension surface in Fig. 10. It is observed that the complexity imposed by our algorithm is always lower than that of VA, which employs the same OF as our algorithm. As can be summarized from Figs. 9 to 10, when the channel SNR and delay are respectively equivalent to 20dB and $\tau = 3/4$ symbol interval and $T = 20$ CLDC codewords are entailed by each ACLDC block, an initial searching breadth of $M = 20$ with an incremental step of $\Delta = 2$ is sufficient to obtain the BER performance of VA with a much lower complexity.

V. Conclusion

In this paper, we have proposed a novel sequential tree-search algorithm to solve the optimization problem posed either by the ACLDC scheme with an asynchronous receiver or by the TR-STBC scheme with frequency selective channels. Theoretical analysis and simulation results have demonstrated that our algorithm is able to remarkably raise the effective symbol rate of the system employed by either the ACLDC or the TR-STBC scheme, with a complexity that is lower than the SD-based algorithms, the VA algorithm and the ML algorithm. This can be achieved as our algorithm not only employs a different objective function with a lower computation complexity than all the previous SD-based algorithms, but also tactfully avoids repetitious calculations. We further extend our algorithm so that the searching breadth can be adjustable based on a prefixed initial value and step, decided according to the channel SNR and delay.

REFERENCES

[1] E. Lindskog and A. Paulraj, "A transmit diversity scheme for channels with intersymbol interference," in *proceedings of the IEEE International Conference on Communications*, New Orleans, LA, Jun. 2000, vol. 1, pp. 307–311.

[2] T. P. Ren, Y. L. Guan, C. Yuen and E. Y. Zhang, "Rate-1 shift group-decodable STBC," in *proceedings of the IEEE International Conference on Information, Communications and Signal Processing*, Macau, Macau, 8-10, Dec. 2009, vol. 1, pp. 1–5.

[3] Z. Li and X.-G. Xia, "A simple Alamouti space-time transmission scheme for asynchronous cooperative systems," *IEEE Signal Process Letter*, vol. 14, no. 11, pp. 804–807, Nov. 2007.

[4] Y. Shang and X.-G. Xia, "Limited-shift-full-rank matrices with applications in asynchronous cooperative communications," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4119–4126, Nov. 2007.

[5] N. Wu and H. Gharavi, "Asynchronous Cooperative MIMO Systems Using a Linear Dispersion Structure," *IEEE Transactions on Vehicular Technology*, vol. 59, No. 2, pp. 779–787, February, 2010.

[6] M. O. Damen and A. R. Hammons,"Delay-tolerant distributed-TAST codes for cooperative diversity,"*IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3755–3773, Oct. 2007.

[7] J. Anderson and S. Mohan, "Sequential coding algorithms: a survey and cost analysis," *IEEE Transactions on Communications*, vol. 32, No. 2, pp. 169–176, February, 1984.

[8] K. J. Kim, J. Yue, R. A. Iltis and J. D. Gibson,"A QRD-M/Kalman filter-based detection and channel estimation algorithm for MIMO-OFDM systems," *IEEE Transactions of Wireless Communications*, vol. 4, pp. 710–721, May 2005.

[9] Y. Kim and K. Cheun, "A reduced-complexity tree-search detection algorithm for MIMO systems,"*IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2420–2424, June, 2009.

[10] T. P. Ren, Y. L. Guan, C. Yuen, and E. Y. Zhang, "Block-orthogonal space-time code structure and its impact on QRD-M decoding complexity reduction," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1438–1450, December, 2011.

[11] K.-C. Lai and L.-W. Lin,"Low-complexity adaptive tree search algorithm for MIMO detection,"*IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp 3716–3726, July, 2009.

[12] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 1566 – 1577, July 2005.

[13] H. Vikalo, B. Hassibi and U. Mitra, "Sphere-constrained ML detection for frequency-selective channels," *IEEE Transactions on Communications*, vol. 54, no. 7, page 1179-1183, Jul 2006.

[14] H. Vikalo and B. Hassibi, "Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding," *EURASIP Journal of Applied Signal Processing*, vol. 2002, no. 5, pp. 525–531, May. 2002.

[15] L. G. Barbero, T. Ratnarajah and C. F. N. Cowan. "On the complexity of the sphere decoder for frequency-selective MIMO channels," *IEEE Transactions on Signal Processing*, vol. 56, No. 12, pp. 6031–6043, Dec., 2008.

[16] J. W. Choi, B. Shim, A. C. Singer and N. I. Cho, "Low-complexity decoding via reduced dimension maximum-likelihood search," *IEEE Transaction on Signal Processing*, vol. 38, no. 3, Mar, 2010.

[17] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1804–824, Jul. 2002.

APPENDIX A

QR-DECOMPOSITION

Now, we will expand our discussion of QR-decomposition in two subsections: Section A.1 - optimization problems in the MIMO system where the detection procedure is provoked per symbol interval; and Section A.2 - optimization problem in our two system models where the detection procedure is activated per block duration comprising T codeword durations and a fixed amount of guard intervals. The detection procedure of a T-element signal vector in a wireless communication system can be regarded as an optimization problem. It can be either solved by a tree search algorithm on an element-by-element basis, whether each element is a symbol or a codeword made up of several symbols; or by a non tree-search algorithm which will obtain all the elements of the output at once or in a non causal, non sequential manner. Below (Section A.1), we provide an example to show how QR-decomposition is employed to transfer an optimization problem that cannot be solved by tree-search algorithms, to an optimization problem that can be solved by tree-search algorithms.

A.1 Optimization problems in the MIMO system with single-block-length transmission

We consider a MIMO system with M_{T} number of transmit antennas and M_{R} number of receive antennas. Flat-fading channels are assumed and no other transceiver scheme is employed. The detection procedure will be activated during each symbol interval, as the system employs a scheme of single-block-length transmission. More exactly, the received signal at any symbol interval can be quantified as:

$$\mathbf{r} = \mathbf{H}\mathbf{b} + \mathbf{n}, \tag{19}$$

where the $(M_{\mathrm{R}} \times 1)$-element vector $\mathbf{n}$ quantifies the additive white Gaussian noise (AWGN) samples encountered at the M_{R} receive antennas. $\mathbf{H}$ is the $(M_{\mathrm{R}} \times M_{\mathrm{T}})$ channel impulse response (CIR) matrix, and $\mathbf{b}$ is the $(M_{\mathrm{T}} \times 1)$ signal vector transmitted from the M_{T} number of antennas. The CIR matrix can

be further expanded as:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M_{\mathrm{T}}} \\ h_{21} & h_{22} & \cdots & h_{2M_{\mathrm{T}}} \\ \vdots & \vdots & \vdots & \vdots \\ h_{M_{\mathrm{R}}1} & h_{M_{\mathrm{R}}2} & \cdots & h_{M_{\mathrm{R}}M_{\mathrm{T}}} \end{bmatrix}, \tag{20}$$

where h_{ij} is the CIR between the ith transmit antenna and the jth receive antenna. The OF represented in the form of Euclidean distance can be quantified as:

$$E(\dot{\mathbf{b}}, \mathbf{r}) = \|\mathbf{r} - \mathbf{H}\dot{\mathbf{b}}\|^2. \tag{21}$$

In order to apply tree-search algorithm, we have to break down the overall OF into M_{R} components, so that:

$$E(\dot{\mathbf{b}}, \mathbf{r}) = \sum_{t=1}^{M_{\mathrm{R}}} e_t\big(\dot{\mathbf{v}}(t), r_t\big), \tag{22}$$

where $e_t(\dot{\mathbf{v}}(t), r_t)$ is the tth local OF. According to (21), $e_t(\dot{\mathbf{v}}(t), r_t)$ can be quantified as:

$$e_t\left(\dot{\mathbf{v}}(t), r_t\right) = \|r_t - \mathbf{h}_t^T\dot{\mathbf{b}}\|^2, \tag{23}$$

where r_t is the tth element of the received signal vector $\mathbf{r}$ and $\mathbf{h}_t^T$ represents the tth row of the CIR matrix $\mathbf{H}$. Obviously, $\dot{\mathbf{v}}(t)$ remains to be the entire trial vector $\dot{\mathbf{b}}$ regardless of the value of t. Hence, we have:

$$\dot{\mathbf{v}}(t) = \dot{\mathbf{b}}, \quad \forall t = 1, 2, \ldots, T. \tag{24}$$

Obviously, with a local OF of (23), the estimation of the entire vector $\dot{\mathbf{b}}$ cannot be pursued on a symbol-by-symbol or codeword-by-codeword basis.

A QR-decomposition is hence introduced to map the detection problem of $\hat{\mathbf{b}}$ into a tree search algorithm. Under such circumstances, the CIR matrix $\mathbf{H}$ has to be QR-decomposed to achieve $\mathbf{H} = \mathbf{QR}$, where the $(M_{\mathrm{R}} \times M_{\mathrm{T}})$ matrix $\mathbf{Q}$ has orthogonal columns, so that $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_{M_{\mathrm{T}}}$ and $\mathbf{I}_{M_{\mathrm{T}}}$ is an $(M_{\mathrm{T}} \times M_{\mathrm{T}})$ identity matrix. The $(M_{\mathrm{T}} \times M_{\mathrm{T}})$-dimensional matrix $\mathbf{R}$ is upper triangular. Therefore, we multiply (19) with $\mathbf{Q}^H$ on both sides yielding;

$$\mathbf{z} = \mathbf{Q}^H\mathbf{r} = \mathbf{Q}^H\mathbf{H}\mathbf{b} + \mathbf{Q}^H\mathbf{n} = \mathbf{Q}^H\mathbf{QRb} + \tilde{\mathbf{n}} = \mathbf{Rb} + \tilde{\mathbf{n}}. \tag{25}$$

Thus, the overall OF can be alternatively defined as:

$$E\big(\dot{\mathbf{b}}, \Omega(\mathbf{r})\big) = E\big(\dot{\mathbf{b}}, \mathbf{z}\big) = \|\mathbf{z} - \mathbf{R}\dot{\mathbf{b}}\|^2. \tag{26}$$

We may write the overall OF in (26) as the sum of M_{T} local OFs, such that

$$E(\dot{\mathbf{b}}, \mathbf{z}) = \sum_{t=1}^{M_{\mathrm{T}}} e_t\big(\dot{\mathbf{v}}(t), z_t\big), \tag{27}$$

with

$$e_t\left(\dot{\mathbf{v}}(t), z_t\right) = \|z_t - \bar{\mathbf{h}}_t^T \dot{\mathbf{b}}\|^2, \tag{28}$$

where z_t is the tth element of $\mathbf{z}$ and $\bar{\mathbf{h}}_t^T$ represents the tth row of the decomposed CIR matrix $\mathbf{R}$, which is a triangular matrix and can be represented as:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1M_{\mathrm{T}}} \\ 0 & R_{22} & R_{23} & \cdots & R_{2M_{\mathrm{T}}} \\ 0 & 0 & R_{33} & \cdots & R_{3M_{\mathrm{T}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{M_{\mathrm{T}}M_{\mathrm{T}}} \end{bmatrix}. \tag{29}$$

Substituting $\bar{\mathbf{h}}_t$ with the tth row of (29), as observed from the triangular feature of $\mathbf{R}$ in (29), (28) can be further derived as:

$$e_t\left(\dot{\mathbf{v}}(t), z_t\right) = \|z_t - \bar{\mathbf{h}}_{t,t\to T}^T \dot{\mathbf{v}}(t)\|^2, \tag{30}$$

where $\bar{\mathbf{h}}_{t,t\to T}^T$ is the row vector containing the last $(M_{\mathrm{T}} - t + 1)$ elements of the tth row of $\mathbf{R}$. Correspondingly, in (30), $\dot{\mathbf{v}}(t)$ can be written as a $(M_{\mathrm{T}} - t + 1)$-element column-vector containing the elements from the tth to the $M_{\mathrm{T}}^{\text{th}}$ element of the entire trial vector $\dot{\mathbf{b}}$. More explicitly, we may have

$$\dot{\mathbf{v}}(t) = [\dot{b}_t \quad \dot{b}_{t+1} \quad \cdots \quad \dot{b}_{M_{\mathrm{T}}}]^T, \quad \forall t = 1, 2, \ldots, M_{\mathrm{T}}. \tag{31}$$

Now combining (30) and (31), we can see that, with the aid of a QR-decomposition the entire trial vector $\dot{\mathbf{b}}$ can be pursued on a symbol-by-symbol basis using a tree-search algorithm on a M_{T}-level tree or a M_{T}-column trellis table, where the states of the tth column employ the tth local OF e_t as shown in (30), with an input of $\dot{\mathbf{v}}(t)$ as shown in (31).

A.2 Optimization problem in our two systems with T-codeword per block

The unified equations of local OF characterizing both optimization problems in our two systems have already been given by (6) in this paper. The exact definition of each parameter in (6) is detailed in the last paragraph of Section II-B, when the time-reversed space-time block code (TR-STBC) transmit scheme is employed in a system having two transmit antennas, one receive antenna, while the channel between each pair of antennas is a $(L+1)$-path frequency selective fading channel. Moreover, the accurate definition of all the parameters in (6) are also detailed in the last paragraph of Section II-C, when a $(K, \bar{T}, Q)$ asynchronous linear dispersive code (ACLDC) scheme is employed in a cooperative transmission having a K-transmit antenna and a single receive antenna and each channel is flat-fading and the 2nd to the Kth antennas are not synchronized with the first antenna. For further explanation about the system parameters K, $\bar{T}$ and Q, interested readers may refer to the original paper.

1) In TR-STBC assisted systems: Without applying QR-decomposition, the overall OF in the TR-STBC assisted frequency-selective systems can be represented as:

$$E\big(\dot{\mathbf{b}}, \Omega(\mathbf{r}), \underline{\mathbf{X}}, \underline{\tilde{\mathbf{X}}}_i, \mathbb{I}\big) = \left\| \mathbf{z} - \underline{\mathbf{X}}\bar{\dot{\mathbf{b}}} - \sum_{i=-L}^{-1} \underline{\tilde{\mathbf{X}}}_i^{(-)} \underline{\tilde{\dot{\mathbf{b}}}}_i - \sum_{i=1}^{L} \underline{\tilde{\mathbf{X}}}_i^{(+)} \underline{\tilde{\dot{\mathbf{b}}}}_i \right\|^2, \tag{32}$$

where the $(2T \times 1)$-element $\mathbf{z}$ can be further expanded as:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_T \end{bmatrix}, \tag{33}$$

where $\mathbf{z}_t = [z_{t1}, z_{t2}]^T$. $\mathbf{z}$ is the resultant vector of $\Omega(\mathbf{r})$. In (32), $\underline{\mathbf{X}}$ is a $(2T \times 2T)$-element second-level diagonal matrix which can be expanded as:

$$\underline{\mathbf{X}} = \begin{bmatrix} \mathbf{X} & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{X} & \cdots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & \mathbf{X} \end{bmatrix} = \begin{bmatrix} X\mathbf{I}_2 & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & X\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & X\mathbf{I}_2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sum_{k=1}^{2}\sum_{i=0}^{L}|h_{ki}|^2}{2(L+1)}\mathbf{I}_2 & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \frac{\sum_{k=1}^{2}\sum_{i=0}^{L}|h_{ki}|^2}{2(L+1)}\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \cdots & \frac{\sum_{k=1}^{2}\sum_{i=0}^{L}|h_{ki}|^2}{2(L+1)}\mathbf{I}_2 \end{bmatrix}, \tag{34}$$

where $\mathbf{0}_{2\times2}$ is a (2×2)-element all-zero matrix. Besides, in (32) the other notations can be further expanded as:

$$\underline{\bar{\dot{\mathbf{b}}}} = \begin{bmatrix} \dot{\mathbf{b}}(1) \\ \dot{\mathbf{b}}(2) \\ \vdots \\ \dot{\mathbf{b}}(T) \end{bmatrix}, \tag{35}$$

where $\dot{\mathbf{b}}(t) = [\dot{b}(t)_1, \dot{b}(t)_2]^T$ are the trial estimation of the two signals making up the tth STBC codeword.

$$\underline{\tilde{\mathbf{X}}}_i^{(-)} = \begin{bmatrix} \tilde{\mathbf{X}}_i^{(-)} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \tilde{\mathbf{X}}_i^{(-)} & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \tilde{\mathbf{X}}_i^{(-)} \end{bmatrix} = \begin{bmatrix} \tilde{X}_i^{(-)}\mathbf{I}_2 & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \tilde{X}_i^{(-)}\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \tilde{X}_i^{(-)}\mathbf{I}_2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sum_{i=-L}^{-1}\sum_{k=1}^{2}\sum_{j=0}^{L+i} h_{k(j-i)}^* h_{kj}}{2(L+1)}\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \cdots & \frac{\sum_{i=-L}^{-1}\sum_{k=1}^{2}\sum_{j=0}^{L+i} h_{k(j-i)}^* h_{kj}}{2(L+1)}\mathbf{I}_2 \end{bmatrix} \tag{36}$$

$$\underline{\tilde{\mathbf{X}}}_i^{(+)} = \begin{bmatrix} \tilde{\mathbf{X}}_i^{(+)} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \tilde{\mathbf{X}}_i^{(+)} & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \tilde{\mathbf{X}}_i^{(+)} \end{bmatrix} = \begin{bmatrix} \tilde{X}_i^{(+)}\mathbf{I}_2 & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \tilde{X}_i^{(+)}\mathbf{I}_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \tilde{X}_i^{(+)}\mathbf{I}_2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sum_{i=1}^{L}\sum_{k=1}^{2}\sum_{j=0}^{L-i} h_{kj}^* h_{k(j+i)}}{2(L+1)}\mathbf{I}_2 & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & & \ddots & \vdots \\ \vdots & & & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \frac{\sum_{i=1}^{L}\sum_{k=1}^{2}\sum_{j=0}^{L-i} h_{kj}^* h_{k(j+i)}}{2(L+1)}\mathbf{I}_2 \end{bmatrix} \tag{37}$$

$$\underline{\tilde{\dot{\mathbf{b}}}}_i = \begin{bmatrix} \dot{\mathbf{b}}(1-i) \\ \dot{\mathbf{b}}(2-i) \\ \vdots \\ \dot{\mathbf{b}}(T-i) \end{bmatrix}, \quad \forall i = -L, \ldots, -1, 1, \ldots, L. \tag{38}$$

As observed from (38), a guard interval covering L STBC codewords has to be inserted before and after the T-codeword signal blocks.

As can be seen, the auxiliary matrices multiplying the trial estimation vector of the transmitted signal are all diagonal and do not require any QR-decomposition. Thus, the overall OF quantified by (32) can be represented as the superposition of T local OFs, as being addressed in (6) in the revised paper. Furthermore, the local OF addressed in the right hand side (RHS) of (6) in the revised paper can be epitomized in the TR-STBC assisted system as:

$$\begin{aligned} e_t\left(\dot{\mathbf{v}}(t), \Omega(\mathbf{r})(t), \mathbf{X}, \tilde{\mathbf{X}}_i, \mathbb{I}\right) &= e_t\left(\dot{\mathbf{v}}(t), \mathbf{z}_t, \mathbf{X}, \tilde{\mathbf{X}}_i^{(+)}, \tilde{\mathbf{X}}_i^{(-)}, \mathbb{I}\right) \\ &= \left\| \mathbf{z}_t - \mathbf{X}\dot{\mathbf{b}}(t) - \sum_{i=-L}^{-1} \tilde{\mathbf{X}}_i^{(-)} \tilde{\dot{\mathbf{b}}}(t-i) - \sum_{i=1}^{L} \tilde{\mathbf{X}}_i^{(+)} \tilde{\dot{\mathbf{b}}}(t-i) \right\|^2. \end{aligned} \tag{39}$$

Through observing (39), we may easily obtain:

$$\dot{\mathbf{v}}(t) = \left[\dot{\mathbf{b}}^T(t-L) \;\; \dot{\mathbf{b}}^T(t-L+1) \;\; \cdots \;\; \dot{\mathbf{b}}^T(t+L-1) \;\; \dot{\mathbf{b}}^T(t+L)\right]^T. \tag{40}$$

Since $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ are both diagonal matrix resulting from the smart design of the TR-STBC scheme, the $(\bar{T} \times T)$-element detection block-output $\hat{\mathbf{B}}$ can be obtained in $\bar{T}$ parallel tree-search detectors, with the local OF corresponding to the tth column of the $\bar{t}^{\text{th}}$ detector quantified as:

$$e_t = \left\| z_{t\bar{t}} - X\dot{b}(t)_{\bar{t}} - \sum_{i=-L}^{-1} \tilde{X}_i^{(-)} \tilde{\dot{b}}(t-i)_{\bar{t}} - \sum_{i=1}^{L} \tilde{X}_i^{(+)} \tilde{\dot{b}}(t-i)_{\bar{t}} \right\|^2, \tag{41}$$

where

$$\dot{\mathbf{v}}(t)_{\bar{t}} = \left[\dot{\mathbf{b}}^T(t-L)_{\bar{t}} \;\; \dot{\mathbf{b}}^T(t-L+1)_{\bar{t}} \;\; \cdots \;\; \dot{\mathbf{b}}^T(t+L-1)_{\bar{t}} \;\; \dot{\mathbf{b}}^T(t+L)_{\bar{t}}\right]^T. \tag{42}$$

Now combining (41) and (42), we can see that the $\bar{t}^{\text{th}}$ row of the entire estimated signal block $\dot{\mathbf{B}} = [\dot{\mathbf{b}}(1) \; \cdots \; \dot{\mathbf{b}}(T)]$ can be obtained on a symbol-by-symbol basis, with (41) quantifying the local OF of the states

in the tth column of the $\bar{t}^{\text{th}}$ detector, and with any state in the tth column of the $\bar{t}^{\text{th}}$ detector representing the estimation for the $\bar{t}^{\text{th}}$ symbols of the $(t-L)^{\text{th}}$, $\cdots$, t^{th}, $\cdots$, $(t+L)^{\text{th}}$ codewords as shown in (42).

2) In ACLDC assisted systems: Without applying QR-decomposition, the overall OF of the optimization problem proposed in the ACLDC scheme assisted system can be quantified as

$$E\big(\dot{\mathbf{B}}, \Omega(\mathbf{r}), \underline{\mathbf{X}}, \underline{\tilde{\mathbf{X}}}_i, \mathbb{I}\big) = \left\| \mathbf{z} - \underline{\mathbf{X}}\bar{\dot{\mathbf{b}}} - \sum_{i=2}^{K} \underline{\tilde{\mathbf{X}}}_i \underline{\tilde{\dot{\mathbf{b}}}}_i \right\|^2, \tag{43}$$

where the $(\bar{T}T \times 1)$-element $\mathbf{z}$ can be further expanded as:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_T \end{bmatrix}, \tag{44}$$

where $\mathbf{z}_t = [z_{t1}, \ldots, z_{t\bar{T}}]^T$. $\underline{\mathbf{X}}$ is a $(\bar{T}T \times QKT)$-element second-level diagonal matrix which can be expanded as:

$$\begin{aligned} \underline{\mathbf{X}} &= \begin{bmatrix} \mathbf{X} & \mathbf{0}_{\bar{T}\times QK} & \cdots & \mathbf{0}_{\bar{T}\times QK} \\ \mathbf{0}_{\bar{T}\times QK} & \mathbf{X} & \cdots & \mathbf{0}_{\bar{T}\times QK} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{\bar{T}\times QK} & \mathbf{0}_{\bar{T}\times QK} & \cdots & \mathbf{X} \end{bmatrix} \\ &= \begin{bmatrix} \big[h_1\mathbf{C}_1, \sqrt{P_2}h_2\mathbf{C}_2, \cdots, \sqrt{P_K}h_K\mathbf{C}_K\big] & \mathbf{0}_{\bar{T}\times QK} & \cdots & \mathbf{0}_{\bar{T}\times QK} \\ \mathbf{0}_{\bar{T}\times QK} & & \ddots & \vdots \\ \vdots & & & \mathbf{0}_{\bar{T}\times QK} \\ \mathbf{0}_{\bar{T}\times QK} & \cdots & \mathbf{0}_{\bar{T}\times QK} & \big[h_1\mathbf{C}_1, \sqrt{P_2}h_2\mathbf{C}_2, \cdots, \sqrt{P_K}h_K\mathbf{C}_K\big] \end{bmatrix} \end{aligned} \tag{45}$$

where $\mathbf{C}_k$ is a $(\bar{T} \times Q)$-element matrix mapping the $(Q \times 1)$-element original signal vector to $\bar{T}$ signals that to be transmitted from the kth antenna and $\mathbf{0}_{\bar{T} \times QK}$ is a $(\bar{T} \times QK)$-element all-zero matrix. Therefore, each $\mathbf{X}$ is a $(\bar{T} \times QK)$-element matrix. The $(QKT \times 1)$-element desired signal vector can be expanded as:

$$\underline{\bar{\dot{\mathbf{b}}}} = \begin{bmatrix} \bar{\dot{\mathbf{b}}}(1) \\ \bar{\dot{\mathbf{b}}}(2) \\ \vdots \\ \bar{\dot{\mathbf{b}}}(T) \end{bmatrix} = \begin{bmatrix} K \text{ times} \left\{ \begin{matrix} \dot{\mathbf{b}}(1) \\ \vdots \\ \dot{\mathbf{b}}(1) \end{matrix} \right. \\ K \text{ times} \left\{ \begin{matrix} \dot{\mathbf{b}}(2) \\ \vdots \\ \dot{\mathbf{b}}(2) \end{matrix} \right. \\ \vdots \\ K \text{ times} \left\{ \begin{matrix} \dot{\mathbf{b}}(T) \\ \vdots \\ \dot{\mathbf{b}}(T) \end{matrix} \right. \end{bmatrix}, \tag{46}$$

where the $(Q \times 1)$-element vector $\dot{\mathbf{b}}(t)$ is the trial estimation of the Q signals making up the tth LDC codeword. The ith $(\bar{T}T \times 2QT)$-element interference auxiliary matrix will be defined as:

$$\underline{\tilde{\mathbf{X}}}_i = \begin{bmatrix} \tilde{\mathbf{X}}_i & \mathbf{0}_{\bar{T}\times 2Q} & \cdots & \mathbf{0}_{\bar{T}\times 2Q} \\ \mathbf{0}_{\bar{T}\times 2Q} & \tilde{\mathbf{X}}_i & \cdots & \mathbf{0}_{\bar{T}\times 2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{\bar{T}\times 2Q} & \mathbf{0}_{\bar{T}\times 2Q} & \cdots & \tilde{\mathbf{X}}_i \end{bmatrix}$$

$$= \begin{bmatrix} \left[\sqrt{P_{i,1}}h_i\mathbf{C}_i, \sqrt{P_{i,2}}h_i\mathbf{C}_i\right] & \mathbf{0}_{\bar{T}\times 2Q} & \cdots & \mathbf{0}_{\bar{T}\times 2Q} \\ \mathbf{0}_{\bar{T}\times 2Q} & & \ddots & \vdots \\ \vdots & & & \mathbf{0}_{\bar{T}\times 2Q} \\ \mathbf{0}_{\bar{T}\times 2Q} & \cdots & \mathbf{0}_{\bar{T}\times 2Q} & \left[\sqrt{P_{i,1}}h_i\mathbf{C}_i, \sqrt{P_{i,2}}h_i\mathbf{C}_i\right] \end{bmatrix}, \quad \forall i = 2, \ldots, K. \tag{47}$$

The $(2QT \times 1)$-element estimation of the signal vector $\underline{\tilde{\dot{\mathbf{b}}}}_i$ that transmitted from the ith asynchronous relay can be represented as:

$$\underline{\tilde{\dot{\mathbf{b}}}}_i = \begin{bmatrix} \tilde{\dot{\mathbf{b}}}_i(1) \\ \tilde{\dot{\mathbf{b}}}_i(2) \\ \vdots \\ \tilde{\dot{\mathbf{b}}}_i(T) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{b}}(0) \\ \dot{\mathbf{b}}(2) \\ \dot{\mathbf{b}}(1) \\ \dot{\mathbf{b}}(3) \\ \vdots \\ \dot{\mathbf{b}}(T-1) \\ \dot{\mathbf{b}}(T+1) \end{bmatrix}, \quad \forall i = 2, \ldots, K. \tag{48}$$

As observed from (48), a guard interval of one LDC codewords has to be inserted between any two adjacent T-codeword signal blocks. As can be seen, the auxiliary matrices $\underline{\mathbf{X}}$ and $\underline{\tilde{\mathbf{X}}}_i$ multiplying the trial

estimation vector of the transmitted signal in (43) do not require any QR-decomposition to be diagonal.

Thus, the overall OF quantified by (43) can be further represented as the sum of T local OFs, as being addressed in (6) in the revised paper. Finally, the local OF addressed in the RHS of (6) in the revised paper can be epitomized in the ACLDC-assisted system as:

$$\begin{aligned} & e_t\left(\dot{\mathbf{v}}(t), \Omega(\mathbf{r})(t), \mathbf{X}, \tilde{\mathbf{X}}_i, \mathbb{I}\right) = e_t\left(\dot{\mathbf{v}}(t), \mathbf{z}_t, \mathbf{X}, \tilde{\mathbf{X}}_i, \mathbb{I}\right) \\ = & \left\| \mathbf{z}_t - \mathbf{X}\bar{\dot{\mathbf{b}}}(t) - \sum_{i=2}^{K} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_i(t) \right\|^2. \end{aligned} \tag{49}$$

Through observing (49), we may easily obtain that:

$$\dot{\mathbf{v}}(t) = \left[\dot{\mathbf{b}}^T(t-1) \;\; \dot{\mathbf{b}}^T(t) \;\; \dot{\mathbf{b}}^T(t+1)\right]^T. \tag{50}$$

Jointly considering (49) and (50) we may see that, without a QR-decomposition, estimation $\dot{\mathbf{B}} = [\dot{\mathbf{b}}(1) \; \cdots \; \dot{\mathbf{b}}(T)]$ can be pursued on a codeword-by-codeword basis on a T-level tree or T-column trellis table, with (49) quantifying the local OF corresponding to the states in the tth column of the trellis table, and with any state in the tth column representing the estimation for the $(t-1)^{\text{th}}$, the tth and the $(t+1)^{\text{th}}$ codeword, as shown in (50).

Appendix B

Proof of Optimization

Before proving Theorem 1, we will firstly propose and prove Lemma 1.

***Lemma** 1:* Based on the algorithm jointly presented by Algs. 1 and 2 is employed with $M = 2^{3Q}$ (for ACLDC) or $M = 2^{2L+1}$ (for TR-STBC), the following equation:

$$\hat{\dot{\mathbf{v}}}_{\gamma}^{(\dot{t})} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}_{\gamma}^{(\dot{t})}} \left\{ \bar{F}_{\dot{t}}(\dot{\mathbf{v}}) \right\}, \tag{51}$$

is established for $\dot{t} = 2, \ldots, T-1$. It can be proved by mathematical induction; the structure of which may be summarized as:

1) (51) is achieved when $\dot{t} = T-1$;
2) If (51) is achieved when $\dot{t} = t$, and it is also satisfied when $\dot{t} = t-1$.

Proof: We will present the proof of (51) following the structure of mathematical induction. We will firstly clarify some notations. $N_{\text{u}} = 2^Q$ (for ACLDC) or $N_{\text{u}} = 2$ (for TR-STBC) denotes the number

of nodes contained by each group in a column of the trellis table. More details about the definition of 'group' can be checked in Section III-C. $N_{\rm b}=3$ (for ACLDC) or $N_{\rm b}=2L+1$ (for TR-STBC) denotes the number of elements contained by the vector representing each node in the trellis table. $N_{\rm g}=2^{2Q}$ (for ACLDC) or $N_{\rm g}=2^{L+1}$ (for TR-STBC) denotes the number of nodes in the first column of the trellis table. Below we starts the proof.

1) When $t=T-1$, there is only one valid path emanating from each node in the $t^{\rm th}$ column. Hence, by assuming $k=\arg\max_{i=1,\ldots,N_{\rm u}}\{f_{T-1}(\mathbf{u}_{\gamma_{(T-1)},i})+f_T(\mathbf{u}_{\gamma_T})\}$, where γ_T is decided by the last $N_{\rm b}-1$ elements of $\mathbf{u}_{\gamma_{(T-1)},i}$, we have $\hat{\bar{\mathbf{v}}}^{(T-1)}_{\gamma_{(T-1)}}=\bar{\mathbf{v}}^{(T-1)}_{\gamma_{(T-1)},k}=\arg\max_{\dot{\mathbf{v}}\in\bar{\mathbb{V}}^{(T-1)}_{\gamma_{T-1}}}\left\{\bar{F}_{T-1}(\dot{\mathbf{v}})\right\}$. That is $\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma}=\arg\max_{\dot{\mathbf{v}}\in\mathbb{V}^{(t)}_{\gamma}}\left\{\bar{F}_t(\dot{\mathbf{v}})\right\}$ is achieved when $t=T-1$.
2) When $\dot{t}=t$, there will be $N_{\rm u}$ nodes $\mathbf{u}_{\gamma_{t-1},i}\in\mathbb{U}_{\gamma_{t-1}}$ with $i=1,\ldots,N_{\rm u}$ in the $(t-1)^{\rm th}$ column. According to the algorithm description

$$q^{(t-1)}_{\gamma_{t-1},i}=f_{t-1}(\mathbf{u}_{\gamma_{t-1},i})+\hat{q}^{(t)}_{\gamma_{t,i}},\quad\forall i=1,\ldots,N_{\rm u} \tag{52}$$

where $\gamma_{t,i}$ is determined by the last $(N_{\rm b}-1)$ bits of $\mathbf{u}_{\gamma_{t-1},i}$, and we have $\gamma_{t,1}\neq\gamma_{t,2}\neq\cdots\neq\gamma_{t,N_{\rm u}}$. Since (51) is maintained when $\dot{t}=t$, we should have

$$\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_{t,i}}=\arg\max_{\dot{\mathbf{v}}\in\mathbb{V}^{(t)}_{\gamma_{t,i}}}\left\{\bar{F}_t(\dot{\mathbf{v}})\right\},\forall i=1,\cdots,N_{\rm u} \tag{53}$$

As given by the definition of $\hat{q}^{(t)}_{\gamma}$ and (53), we should have $\hat{q}^{(t)}_{\gamma_{t,i}}=\max_{\dot{\mathbf{v}}\in\mathbb{V}^{(t)}_{\gamma_{t,i}}}\left\{\bar{F}_t(\dot{\mathbf{v}})\right\}$, $\forall,i=1,\ldots,N_{\rm u}$. Additionally as given by Line 15 of Alg. 1, we may define set $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$ as the set containing all the vectors having all elements being zero except the $N_{\rm b}$ elements, i.e. from the $(t-1-(N_{\rm b}-1)/2)^{\rm th}$ to the $(t-1+(N_{\rm b}-1)/2)^{\rm th}$ elements equaling to $\mathbf{u}_{\gamma_{t-1},i}$. Therefore, the regional elite partial vector $\hat{\bar{\mathbf{v}}}^{(t-1)}_{\gamma_{t-1},i}$ having the highest accumulated fitness value in $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$ should be $\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_{t,i}}$, with its $(t-1-(N_{\rm b}-1)/2)^{\rm th}$ element changed to $u_{\gamma_{t-1},i,1}$, yielding $\bar{\mathbf{v}}^{(t)}_{\gamma_{t,i}}$. That is: $\bar{\mathbf{v}}^{(t-1)}_{\gamma_{t-1},i}=\arg\max_{\dot{\mathbf{v}}\in\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}}\{\bar{F}_{t-1}(\dot{\mathbf{v}})\},\forall i=1,\ldots,N_{\rm u}$. According to the definition of $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$, the $(t-1-(N_{\rm b}-1)/2)^{\rm th}$, $\cdots$, $(t-1+(N_{\rm b}-1)/2)^{\rm th}$ elements of all the vectors belonging to $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$ are the same. Hence, $f_{t-1}(\dot{\mathbf{v}})$ is the same for any $\dot{\mathbf{v}}\in\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$. Therefore, following (52) we have

$$q^{(t-1)}_{\gamma_{t-1},i}=\max_{\dot{\mathbf{v}}\in\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}}\{\bar{F}_{t-1}(\dot{\mathbf{v}})\},\forall i=1,\cdots,N_{\rm u}. \tag{54}$$

According to (54) and the definition of $\hat{q}_{\gamma_{t-1}}^{(t-1)}$, we have

$$\hat{q}_{\gamma_{t-1}}^{(t-1)} = \max_{i=1,\cdots,N_{\mathrm{u}}} \left\{ q_{\gamma_{t-1},i}^{(t-1)} \right\} = \max_{i=1,\cdots,N_{\mathrm{u}}} \left\{ \max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}_{\gamma_{t-1},i}^{(t-1)}} \{ \bar{F}_{t-1}(\dot{\mathbf{v}}) \} \right\}. \tag{55}$$

Since $\bar{\mathbb{V}}_{\gamma_{t-1}}^{(t-1)} = \bigcup_{i=1}^{N_{\mathrm{u}}} \bar{\mathbb{V}}_{\gamma_{t-1},i}^{(t-1)}$, (55) can be further simplified as: $\hat{q}_{\gamma_{t-1}}^{(t-1)} = \max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}_{\gamma_{t-1}}^{(t-1)}} \{ \bar{F}_{t-1}(\dot{\mathbf{v}}) \}$. Therefore, $\hat{\dot{\mathbf{v}}}_{\gamma_{t-1}}^{(t-1)} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}_{\gamma_{t-1}}^{(t-1)}} \left\{ \bar{F}_{t-1}(\dot{\mathbf{v}}) \right\}$. So far, we have proved that as long as (51) exists for $\dot{t} = t$, it will also exist for $\dot{t} = t-1$.

■

Since we have proved Lemma 1, we will begin to prove Theorem 1.

Proof: Based on the above two items, (51) will always be true within the range of $\dot{t} = 2, \ldots, T-1$. We will further prove $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}} \left\{ F\left(\dot{\mathbf{v}}\right) \right\}$. For any given node $\mathbf{u}_\gamma$ in the first column, we define the set entailing vectors starting with $\mathbf{u}_\gamma$ as set $\mathbb{V}_\gamma^{(1)}$. Apparently, we have $\mathbb{V}_\gamma^{(1)} = \bar{\mathbb{V}}_\gamma^{(2)}$. As further derived from (51), we have

$$\hat{\dot{\mathbf{v}}}_\gamma^{(2)} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(2)}} \left\{ \bar{F}_2(\dot{\mathbf{v}}) \right\}, \forall \gamma = 1 \cdots N_{\mathrm{g}}. \tag{56}$$

Since all vectors in $\mathbb{V}_\gamma^{(1)}$ have the same first $(N_{\mathrm{b}}+1)/2$ elements, $f_1(\dot{\mathbf{v}})$ will be the same for any vector $\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(1)}$. Therefore, as suggested by Line 11 of Alg. 2 and (56),

$$q_\gamma^{(1)} = \max_{\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(1)}} \left\{ f_1\left(\dot{\mathbf{v}}\right) + \bar{F}_2\left(\dot{\mathbf{v}}\right) \right\} = \max_{\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(1)}} \left\{ \bar{F}_1\left(\dot{\mathbf{v}}\right) \right\} = \max_{\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(1)}} \left\{ F\left(\dot{\mathbf{v}}\right) \right\}, \forall \gamma = 1 \cdots N_{\mathrm{g}} \tag{57}$$

According to Lines 12 and 13 in Alg. 2 and (57), we will have

$$\hat{\mathbf{v}} = \arg\max_{\gamma=1\cdots N_{\mathrm{g}}} \left\{ q_\gamma^{(1)} \right\} = \arg\max_{\gamma=1\cdots N_{\mathrm{g}}} \left\{ \max_{\dot{\mathbf{v}} \in \mathbb{V}_\gamma^{(1)}} \left\{ F\left(\dot{\mathbf{v}}\right) \right\} \right\}. \tag{58}$$

We have $\mathbb{V} = \bigcup_{\gamma=1}^{N_{\mathrm{g}}} \mathbb{V}_\gamma^{(1)}$ and (58) can be further simplified as: $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}} \left\{ F\left(\dot{\mathbf{v}}\right) \right\}$. Thus, we have proved Theorem 1, i.e. the output given by our algorithm BATSA, when $M = 2^{3Q}$ (for ACLDC) or $M = 2^{2L+1}$ (for TR-STBC) will always be the optimal solution. ■

Algorithm 1: Recursive function **SearchRt** searching for leaf-nodes with $t = 2 \cdots T$.

input : t, γ_t

output: p_t, $\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_t}$, $\hat{q}^{(t)}_{\gamma_t}$

$i \leftarrow 0$; $p_t \leftarrow 0$

for $j \leftarrow 1$ **to** M **do**

  if $\mathbf{u}_r \in \mathbb{U}_{\gamma_t}$ **then**

    $i \leftarrow i + 1$

    if $t = T$ **then**

      $\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_t} \leftarrow [\underbrace{0, \ \cdots, \ 0}_{T-2}, \ u_{j1}, \ \cdots, \ u_{j\frac{N_\mathrm{b}+1}{2}}]$

      $\hat{q}^{(t)}_{\gamma_t} \leftarrow f_T(\mathbf{u}_j, \mathbf{z}_t)$

      $p_t \leftarrow 1$; **return**

    else

      $\gamma_{t+1} \xleftarrow{\text{decide}} [u_{j2}, \ \cdots, \ u_{jN_\mathrm{b}}]$

      if $\hat{q}^{(t+1)}_{\gamma_{t+1}}$ *has NOT been calculated* **then** $\left\{p_{t+1}, \hat{\bar{\mathbf{v}}}^{(t+1)}_{\gamma_{t+1}}, \hat{q}^{(t+1)}_{\gamma_{t+1}}\right\} \leftarrow$ **SearchRt**$(t+1, \gamma_{t+1})$

      else $p_{t+1} \leftarrow 1$

      if $p_{t+1} = 1$ **then**

        $\boxed{\text{Straight Top Break}}$

        $\bar{\mathbf{v}}^{(t)}_{\gamma_t,i} \leftarrow [\underbrace{0, \ \cdots, \ 0}_{t-2}, \ u_{j1}, \ \hat{\bar{v}}^{(t+1)}_{\gamma_{t+1}t}, \ \cdots, \ \hat{\bar{v}}^{(t+1)}_{\gamma_{t+1}T}]$

        $q^{(t)}_{\gamma_t,i} \leftarrow f_t(\mathbf{u}_j, \mathbf{z}_t) + \hat{q}^{(t+1)}_{\gamma_{t+1}}$

$k \leftarrow \arg\max_{i=1,\cdots,N_\mathrm{g}} \left\{q^{(t)}_{\gamma_t,i}\right\}$

$\hat{q}^{(t)}_{\gamma_t} \leftarrow q^{(t)}_{\gamma_t,k}$

$\hat{\bar{\mathbf{v}}}^{(t)}_{\gamma_t} \leftarrow \bar{\mathbf{v}}^{(t)}_{\gamma_t,k}$

$p_t \leftarrow 1$; **return**

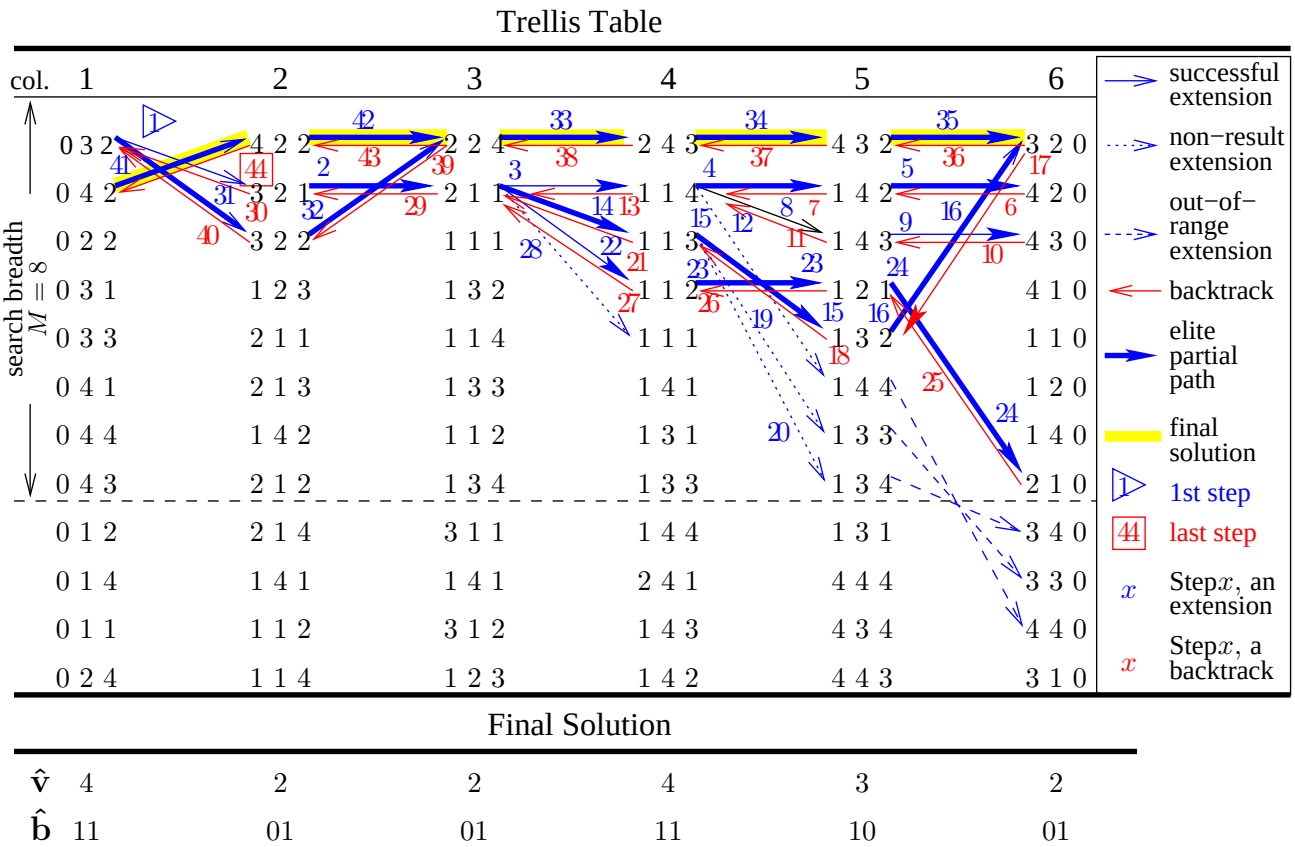

Fig. 1: An example of our decoding algorithm manipulated on a trellis table with $M = 8$ and $T = 6$. The number, regardless of whether in blue or red, indicates the sequence number of the manipulating steps. On one hand, the blue numbers label all the exploring operations. On the other hand, the red numbers refer to the backtracking process. The path highlighted in bold yellow color represents the final solution obtained by the tree-search algorithm, as in the given example.

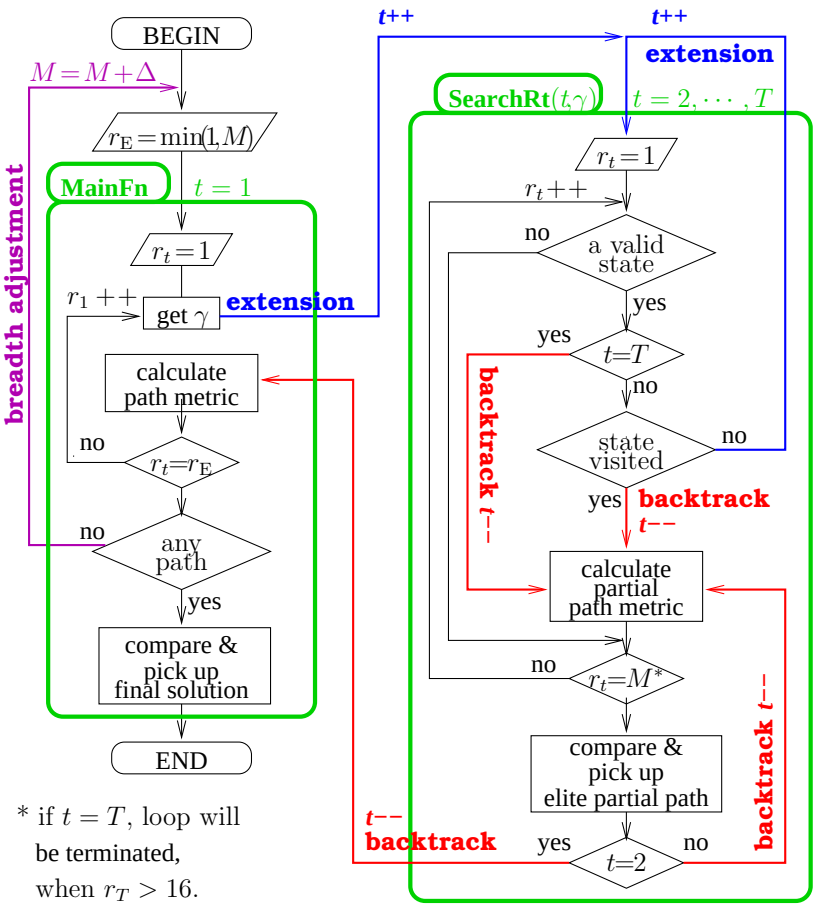

Fig. 2: Flow-chart of the breadth-adjustable tree search algorithm. The detailed pseudo-code of both **MainFn** and **SearchRt** are presented in Algs. 2 and 1 respectively. Besides, the move of 'extension' and 'backtrack' are marked with the same color in the example shown in Fig. 1.

TABLE II: Complexity imposed by calculating the local OFs of all the states in the trellis table employed by the SDAs and BATSA, quantified in real-number FLOPS.

	ACLDC		TR-STBC	
Components	BATSA	SDAs	BATSA	SDAs
local OF	(16)	(14)	(17)	(15)
$\mathbf{a}$ [a]	$16N_{\mathrm{r}}$ [b]	$16N_{\mathrm{r}}$ [b]	$2N_{\mathrm{r}}L$ [b]	$2N_{\mathrm{r}}L$ [b]
$\mathbf{a}^H\mathbf{a}$	$3\bar{T}N_{\mathrm{r}}$	0	$3N_{\mathrm{r}}$	0
$\Re\{\mathbf{a}^H\mathbf{z}\}$	$(3\bar{T}+\bar{T}-1)N_{\mathrm{r}}T$	0	$3N_{\mathrm{r}}T\bar{T}$	0
$\mathbf{d}=\mathbf{z}_t-\mathbf{a}$	0	$2\bar{T}N_{\mathrm{r}}T$	0	$2N_{\mathrm{r}}T\bar{T}$
$\|\mathbf{d}\|^2$	0	$(3\bar{T}+\bar{T}-1)N_{\mathrm{r}}T$	0	$3N_{\mathrm{r}}T\bar{T}$
Overall Difference SDAs $-$ BATSA	$2\bar{T}N_{\mathrm{r}}T-3\bar{T}N_{\mathrm{r}}$		$2N_{\mathrm{r}}T\bar{T}-3N_{\mathrm{r}}$	
Condition for SDAs $>$ BATSA	$T>\frac{3}{2}$		$\bar{T}T>\frac{3}{2}$	

[a] $\mathbf{a}=\mathbf{X}\bar{\dot{\mathbf{b}}}_j+\sum_{i\in\mathbb{I}}\tilde{\mathbf{X}}_i\tilde{\dot{\mathbf{b}}}_j$

[b] BPSK modulation is assumed, complexity imposed by calculating $\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ is omitted, as they are required only once during the detection intercourse, and will not affect the result of the complexity difference between BATSA and SDAs.

TABLE III: Difference between BATSA and SDAs

Components	BATSA	SD	VA/SD
Local OF	(16) ⟨(17)⟩	(14) ⟨(15)⟩	(14) ⟨(15)⟩
Procedures where the accumulated fitness value is calculated	Backtracking	Forward extension	Forward extension & backtracking
Usage of Markovian Property	Yes	No	Yes
STB complexity reduction	Yes	No	No
Breadth Adjustability	Yes	No	No

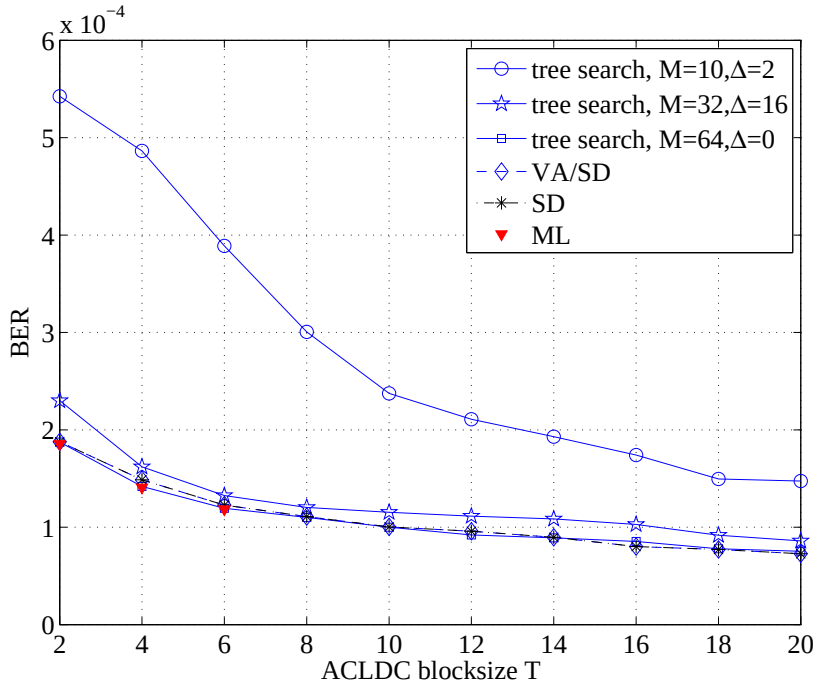

Fig. 3: BER performance versus the ACLDC block size T, when different detection algorithms are applied at system SNR= 20dB, in combination with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$ and delay $\tau = 3/4$. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

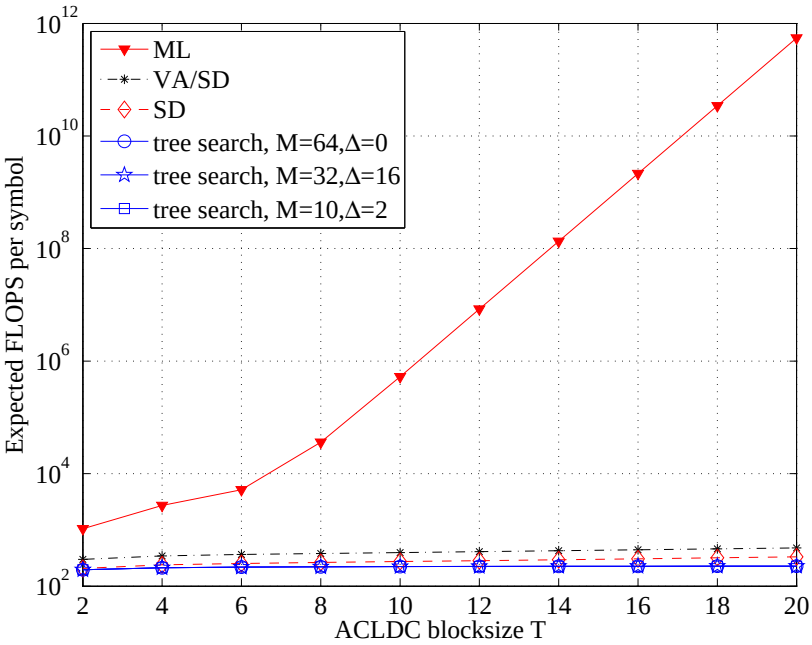

Fig. 4: Complexity versus the ACLDC block size T, when different detection algorithms are applied at system SNR$=20$dB, in combination with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$ and delay $\tau = 3/4$. The complexity of all tree-search algorithms are obtained through simulation, which is calculated as the mean value of the overall number of additions, multiplications and comparisons required for decoding an ACLDC block divided by the number of information symbols $2T$, averaged through 100,000 number of blocks. .

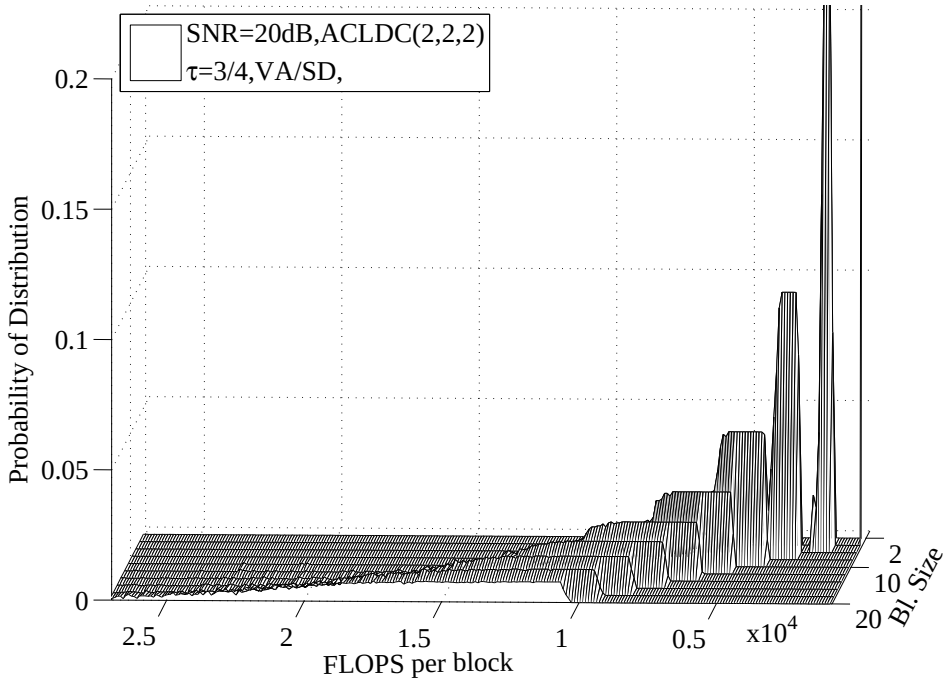

Fig. 5: Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial block size T, when VA combined SD algorithm is applied. All the point in this figure is obtained at system SNR$=20$dB in conjunction with the ACLDC parameters $K=2$, $\bar{T}=2$, $Q=2$. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

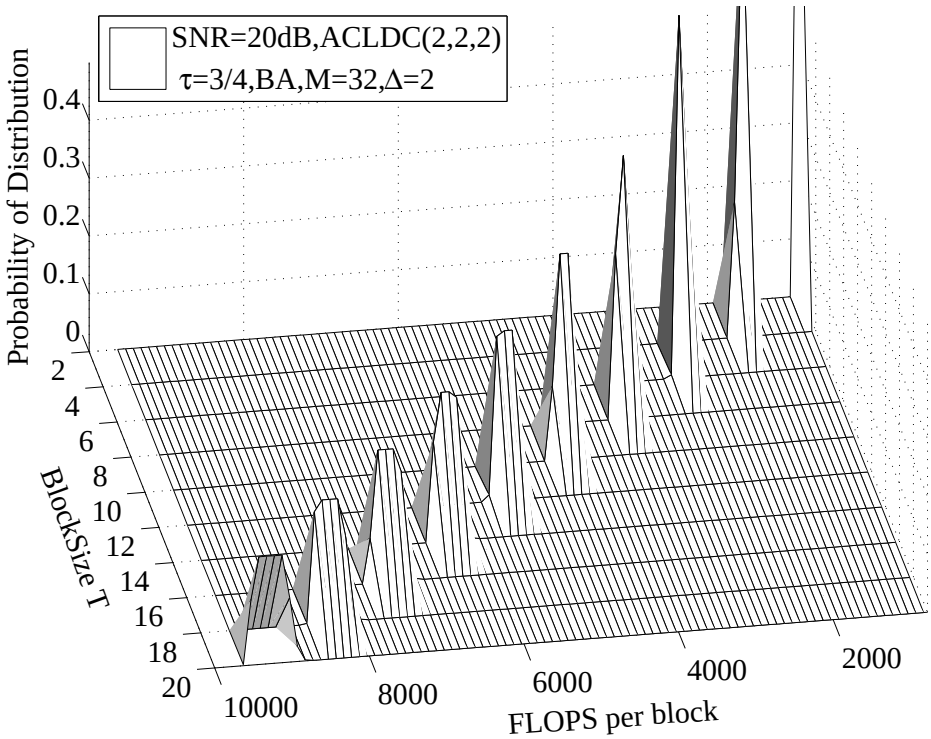

Fig. 6: Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial block size T, when our algorithm is applied with an initial searching breadth of $M = 32$ and an incremental searching step of $\Delta = 2$. All the point in this figure is obtained at system SNR$=$ 20dB in conjunction with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

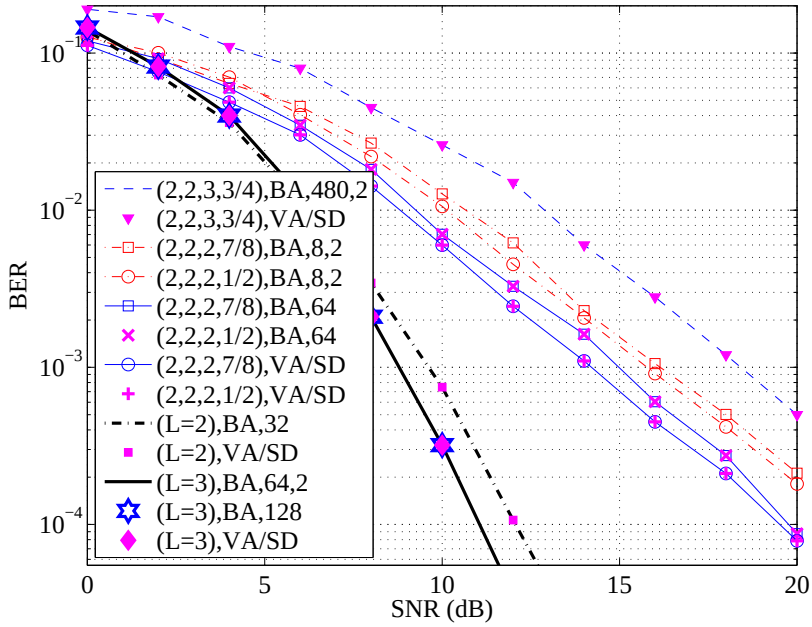

Fig. 7: BER performance versus the system SNR, when different detection algorithms are applied in systems employing an ACLDC scheme associated with $T = 20$ or an TR-STBC with $T = 8$ codewords per block, while the frequency selective channel has $L = 3$ or $L = 2$ delay paths. The numbers in the brackets that start with '2,2,2' or '2,2,3' denotes an ACLDC system with $K = 2$, $\bar{T} = 2$, $Q = 2$ or $Q = 2$. The fourth number in the brackets quantifies the delay of the second antenna in an ACLDC system. The legends starting with '$(L = 2)$' or '$(L = 3)$' denotes the 3-path or 4-path TR-STBC system. The first scalar following 'BA' represents the initial searching breadth and the second scalar quantifies the incremental step of the searching breadth, if the initial searching breadth does not equal to the number of rows contained by a column in the entire trellis table. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

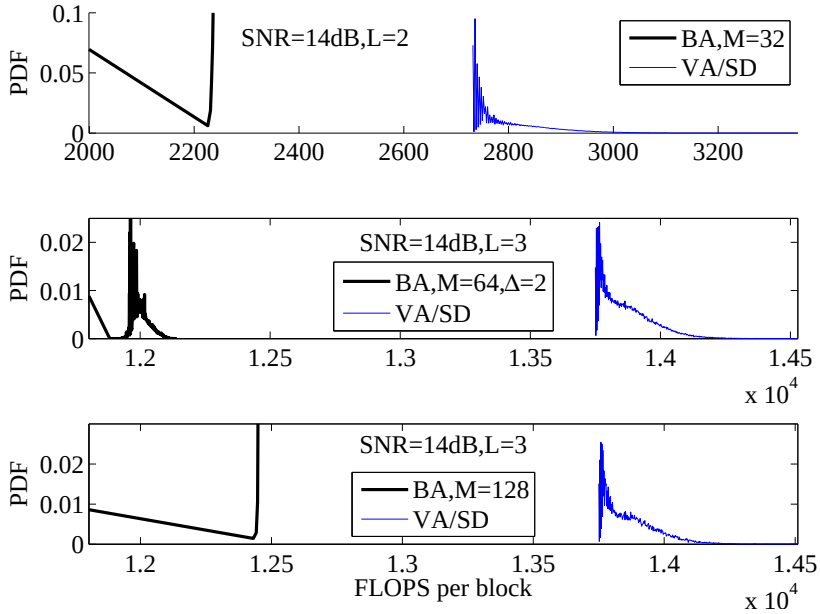

Fig. 8: Comparison of PDF of FLOPS per block of the VA/SD algorithm and our algorithm in TR-STBC assisted system obtained at SNR= 14dB, while the frequency selective channel has 3 or 4 paths and each block contains $T = 8$ codewords. A searching breadth of $M = 64$ or $M = 128$ is employed when TR-STBC is employed in a system with four-path frequency selective channels. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

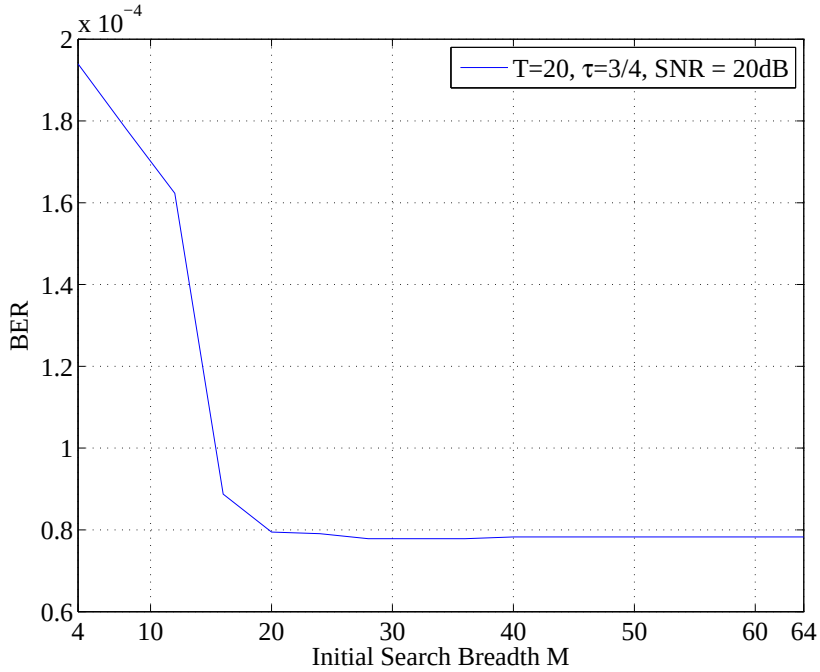

Fig. 9: BER performance versus various initial searching breadth M at a given $\Delta = 2$, when an ACLDC block is made up of $T = 20$ CLDC codewords and one guard interval as the delay of the asynchronous node is $\tau = 3/4$. Additionally, all the different detection algorithms are applied at system SNR= 20dB, in combination with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

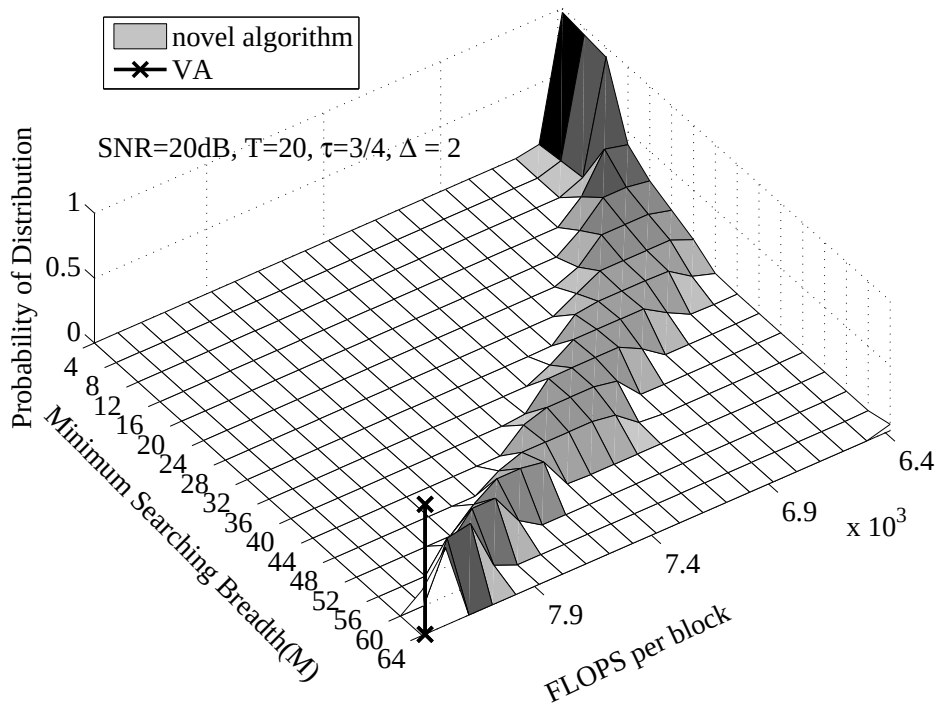

Fig. 10: Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial searching breadth M, when incremental step is $\Delta = 2$. One ACLDC block is made up of $T = 20$ CLDC codewords and one guard interval. All the point in this figure is obtained at system SNR$=$ 20dB in conjunction with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$. The VA employs the same OF as employed by our algorithm BATSA. The total number of ACLDC blocks used for simulation is sufficient to detect 100 erroneous bits.

Algorithm 2: Main intercourse of the proposed algorithm, including the main function - **MainFn**.

input : M, Δ, fitness table

output: $\hat{\mathbf{v}}$

```
n ← 0
while n = 0 do
    M ← M + Δ
    for j ← 1 to M do
        γ ←decide [u_{j2}, ··· , u_{jN_b}]
        {p, v̂̄^(2)_γ, q̂^(2)_γ} ← SearchRt(2, γ)
        if p = 1 then
            n = n + p
            [Straight Top Break]
            v̄^(1)_n ← v̂̄^(2)_γ
            q^(1)_n ← f_1(u_{j_1}, z_1) + q̂^(2)_γ
k ← arg max_{i=1···n} {q^(1)_i}
v̂ ← v̄^(1)_k
```