A Low-Complexity Solution to Decode Diversity-Oriented Block Codes in MIMO Systems with Inter-Symbol Interference

Chong Xu and Hamid Gharavi

***Abstract*—In this paper we first 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 simulations 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.**

***Index Terms*—Multi Input Multi Output (MIMO), Inter-symbol Interference (ISI), Time-Reversed Space-Time Block Code (TR-STBC), Linear Dispersion Code (LDC), asynchronous cooperative transmission, tree-search detection algorithm, sphere decoding algorithm, Viterbi Algorithm (VA), Maximum Likelihood (ML) algorithm.**

I. Introduction

INTER-SYMBOL Interference (ISI) can be caused by many factors. 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 $T_{\rm s}$ [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 extensive [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

Manuscript received October 5, 2011; revised March 2 and May 15, 2012; accepted July 3, 2012. The associate editor coordinating the review of this paper and approving it for publication was M. Ardakani.

The authors are with the National Institute of Standards and Technology, Gaithersburg, USA (e-mail: {chong.xu, hamid.gharavi}@nist.gov).

Digital Object Identifier 10.1109/TWC.2012.081612.111819

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 a 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:

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

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 k^{th} 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: $\bar{\mathbf{S}}_{\bar{t}} = \left[\ \mathbf{s}_{\bar{t}}^{(1)} \quad \mathbf{s}_{\bar{t}}^{(2)} \quad \cdots \quad \mathbf{s}_{\bar{t}}^{(T)}\ \right]$, $\forall \bar{t} = 1, \ldots, \bar{T}$. $\bar{\mathbf{S}}_{\bar{t}}$ 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 i^{th} ISI component contaminating the signals received during the $\bar{t}^{\text{th}}$ phase.

The t^{th} 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:

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

$\mathbf{X}$ and $\tilde{\mathbf{X}}_i$ respectively denote the auxiliary matrix associated with the desired signal vector $\bar{\mathbf{b}}_t$, and the i^{th} ISI component $\tilde{\mathbf{b}}_{ti}$, with respect to the t^{th} decision input vector $\mathbf{z}_t$. As observed from (4), $\mathbf{z}_t$ can be regarded as the t^{th} 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:

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

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 $\underline{\dot{\mathbf{b}}}$ and of the i^{th} ISI component $\underline{\tilde{\mathbf{b}}}_i$, when the entire block is considered. More exactly, we have $\underline{\mathbf{X}} = \mathbf{X} \otimes \mathbf{I}_T$, $\underline{\tilde{\mathbf{X}}}_i = \tilde{\mathbf{X}}_i \otimes \mathbf{I}_T$, $\underline{\mathbf{z}} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_T^T]^T$, $\underline{\bar{\dot{\mathbf{b}}}} = [\bar{\dot{\mathbf{b}}}^T(1), \bar{\dot{\mathbf{b}}}^T(2), \ldots, \bar{\dot{\mathbf{b}}}^T(T)]^T$, $\underline{\tilde{\mathbf{b}}}_i = [\tilde{\mathbf{b}}_i^T(1), \tilde{\mathbf{b}}_i^T(2), \ldots, \tilde{\mathbf{b}}_i^T(T)]^T$, 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 t^{th} 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 t^{th} 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

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

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 a report as the extended version of this paper in [17]. 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:

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

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

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

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 (10), where the power of the transmitted signal is assumed to be uniformly distributed among the $(L+1)$ frequency-selective sub-channels, hence the power of the signal associated with each sub-channel is assumed to be $\rho = 1/\sqrt{L+1}$. 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 i^{th} 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 $\mathbf{A} = 1/\sqrt{2}\rho\mathbf{I}_2$ and $\tilde{\mathbf{A}}_i = 1/\sqrt{2}\rho\mathbf{I}_2$ 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:

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} z_{11} & \cdots & z_{t1} & \cdots & z_{T1} \\ z_{12} & \cdots & z_{t2} & \cdots & z_{T2} \end{bmatrix} \\ &= 1/\sqrt{2}\rho\mathbf{H}^H\bar{\mathbf{r}} = 1/\sqrt{2}\rho \begin{bmatrix} h_1^*(q) & h_2(q^{-1}) \\ h_2^*(q) & -h_1(q^{-1}) \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \bar{\mathbf{r}}_2 \end{bmatrix} \\ \mathbf{z}_t &= 1/\sqrt{2}\rho \begin{bmatrix} \sum_{i=0}^{L} h_{1i}^* r_{(t+i)1} + \sum_{i=0}^{L} h_{2i}\bar{r}_{(t-i)2} \\ \sum_{i=0}^{L} h_{2i}^* r_{(t+i)1} - \sum_{i=0}^{L} h_{1i}\bar{r}_{(t-i)2} \end{bmatrix}. \end{aligned} \tag{11}$$

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.

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

More exactly, as for the desired signal, we have $\bar{\mathbf{b}}_t = [b_{t1},\ b_{t2}]^T$ and $\mathbf{X} = 0.5\rho^2 \sum_{k=1}^{2}\sum_{i=0}^{L}|h_{ki}|^2\mathbf{I}_2$, 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 $\tilde{\mathbf{X}}_i = 0.5\rho^2 \sum_{i=1}^{L}\sum_{k=1}^{2}\sum_{j=0}^{L-i} h^*_{kj}h_{k(j+i)}\mathbf{I}_2$, when $i = 1,\ldots,L$; and $\tilde{\mathbf{X}}_i = 0.5\rho^2 \sum_{i=-L}^{-1}\sum_{k=1}^{2}\sum_{j=0}^{L+i} h^*_{k(j-i)}h_{kj}\mathbf{I}_2$, 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 $\bar{n}_{t1} = 0.5\rho^2\Big(\sum_{i=0}^{L} h^*_{1i}n_{(t+i)1} + \sum_{i=0}^{L} h_{2i}n^*_{(T+1-t+i)2}\Big)$ and $\bar{n}_{t2} = \rho\Big(\sum_{i=0}^{L} h^*_{2i}n_{(t+i)1} - \sum_{i=0}^{L} h_{1i}n^*_{(T+1-t+i)2}\Big)$. 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 k^{th} 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 [18] 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: $\mathbf{S}_t = \Lambda(\mathbf{b}_t) = \bar{\mathbf{C}}\mathbf{B}_{\underline{t}}$, 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 k^{th} 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 i^{th} 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 t^{th} 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 i^{th} $(2Q\times 1)$-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 $N_c = 2^{QT}$ (for ACLDC) and $N_c = 2^T$ (for TR-STBC). We may index the candidate solution as $\dot{\mathbf{v}}_i$, with $i = 1, 2, \ldots, N_c$ 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 j^{th} row of the t^{th} column can be denoted as (j,t) and the input trial vector of it can be represented by a N_b-element vector: $\mathbf{u}_j = [u_{j,1}, u_{j,2}, \ldots, u_{j,N_b}]$, while $N_b = 3$ (for ACLDC) or $N_b = 2L+1$ (for TR-STBC). The implication of $\mathbf{u}_{j_t}$ can be interpreted by the elements of $\dot{\mathbf{v}}_i$ as (13) where j_t is a real number ranging between $[1, N_r]$ and $N_r = 2^{3Q}$ (for ACLDC) or $N_r = 2^{2L+1}$ (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 $[u_{j_t,2}, \cdots, u_{j_t,N_b}] = [u_{j_{t+1},1}, \cdots, u_{j_{t+1},N_b-1}]$.

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:

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

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), $\bar{\mathbf{b}}_j = [\mathcal{F}^T(u_{j,2}), \cdots, \mathcal{F}^T(u_{j,2})]^T$ with $\mathcal{F}(u_{j,2})$ repeated K times, and $\mathcal{F}$ 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), $\tilde{\bar{\mathbf{b}}}_j = [\mathcal{F}^T(u_{j,1}), \mathcal{F}^T(u_{j,3})]^T$. 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:

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

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 $\tilde{x}_i = 0.5\rho^2 \sum_{i=1}^{L} \sum_{k=1}^{2} \sum_{j=0}^{L-i} h_{kj}^* h_{k(j+i)}$, when $i = 1, \ldots, L$; and $\tilde{x}_i = 0.5\rho^2 \sum_{i=-L}^{-1} \sum_{k=1}^{2} \sum_{j=0}^{L+i} h_{k(j-i)}^* h_{kj}$, when $i = -L, \ldots, -1$. Besides, in (17) $\bar{b}_j = \mathcal{F}(u_{j,L+1})$ and $\tilde{\bar{b}}_{j,i} = \mathcal{F}(u_{j,L+1-i})$, for all $i = -L, \ldots, -1$ and $i = 1, \ldots, L$, where $\mathcal{F}$ 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: $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_c} \min \sum_{t=1}^{T} e_t(\mathbf{u}_{j_t}, \mathbf{z}_t)$ in ACLDC system or $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_c} \min \sum_{t=1}^{T} e_t(\mathbf{u}_{j_t}, z_{t\bar{t}})$ 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 $\bar{\mathbf{b}}(t)$ and $\dot{\mathbf{b}}(t,i)$. More exactly, in the ACLDC system, the local objective function (OF) of node $\mathbf{u}_j$ in the t^{th} column of hte trelis table employed by our decoding algorithm BATSA can be quantified as (16). [!t] The local OF of node (j,t) employed by our BATSA-based detector in a TR-STBC assisted system can be quantified as (17). [!t] Then the optimal solution can be obtained by solving: $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_c} \max \sum_{t=1}^{T} f_t(\mathbf{u}_{j_t}, \mathbf{z}_t)$, in an ACLDC assisted system or $\hat{\mathbf{v}} = \arg_{\dot{\mathbf{v}}_i \in \mathbb{V}, i=1,\ldots,N_c} \max \sum_{t=1}^{T} f_t(\mathbf{u}_{j_t}, z_{t\bar{t}})$ 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 $N_r(t)$ nodes within the t^{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. 1. 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. 1, starts from the 1^{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 2^{nd} 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. 1, will be operated in turn at any search-depth from the $(T-1)^{\text{th}}$ column backwards to the 2^{nd} 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.

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

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

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

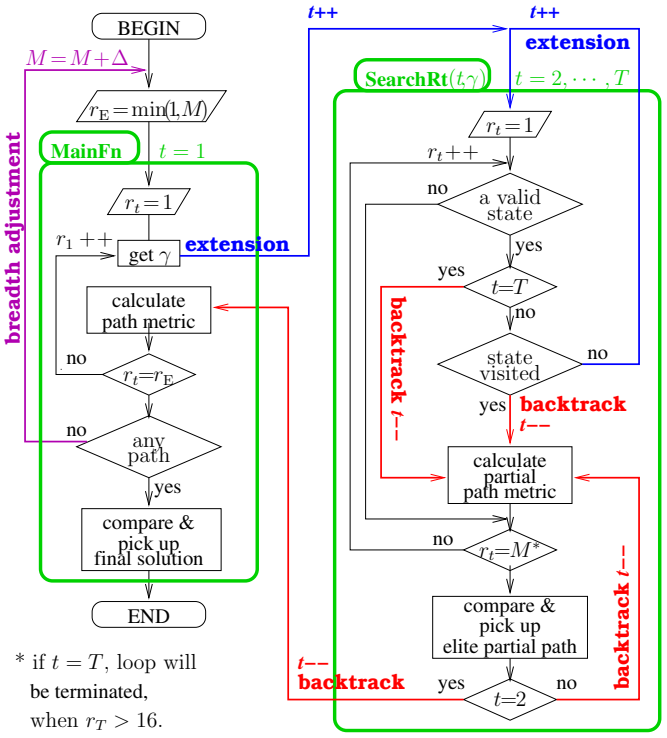

Fig. 1. Flow-chart of the breadth-adjustable tree search algorithm. The detailed pseudo-code of both **MainFn** and **SearchRt** are presented in Algs. 2 and 1 respectively. Besides, the move of 'extension' and 'backtrack' are marked with the same color in the example shown in Fig. 2.

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 t^{th} 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 γ^{th} subset is denoted as $\mathbb{V}_\gamma^{(t)}$, and $\mathbb{V} = \bigcup_{\gamma=1}^{N_g} \mathbb{V}_\gamma^{(t)}$, with $N_g = 2^{2Q}\ \langle N_g = 2^{2L}\rangle$. 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 t^{th} column of the trellis table can be divided into $N_g = 2^{2Q}\ \langle 2^{2L}\rangle$ groups, according to the value of their first 2 $\langle 2L\rangle$ elements. Obviously, all the nodes within the same group in the t^{th} 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 $\gamma_t \in [1, N_g]$ and $i_t \in [1, N_u]$, with $N_u = 2^Q\ \langle 2\rangle$. There will be a unique pair $\{\gamma_t, i_t\}$ corresponding to each $j = 1, \ldots, N_r(t)$.

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 $\bar{F}_t(\dot{\mathbf{v}}_i) = \sum_{t=t}^{T} f_t(\mathbf{u}_{j_t})$, 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 t^{th} 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 T^{th} 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 1^{st} to the T^{th} column has been established;
2) If the current column is not the T^{th}, 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 $\hat{\bar{\mathbf{v}}}_{\gamma_{t+1}}^{(t+1)}$ from the $(t+1)^{\text{th}}$ column, affix it with the first digit of the current node, i.e. $u_{\gamma_t,i_t,1}^{(t)}$, and copy the elements from the t^{th} to the T^{th} position of $\hat{\bar{\mathbf{v}}}_{\gamma_{t+1}}^{(t+1)}$ to $\bar{\mathbf{v}}_{\gamma_t,i_t}^{(t)}$;
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 $\hat{q}_{\gamma_{t+1}}^{(t+1)}$ 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 $q_{\gamma_t,i_t}^{(t)}$;
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 t^{th} 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 $q_{\gamma_t,\bar{i}}^{(t)}$ through $\bar{i} = 1, \ldots, N_u$. Return to the $(t-1)^{\text{th}}$ column with the vector $\bar{\mathbf{v}}_{\gamma_t,k}^{(t)}$ having the highest $q_{\gamma_t,k}^{(t)}$ as $\hat{\bar{\mathbf{v}}}_{\gamma_t}^{(t)}$ in combination with $q_{\gamma_t,k}^{(t)}$ as $\hat{q}_{\gamma_t}^{(t)}$.

Eventually, the final solution $\hat{\mathbf{v}}$ is equivalent to $\hat{\bar{\mathbf{v}}}_{\hat{\gamma}_1}^{(1)}$, where

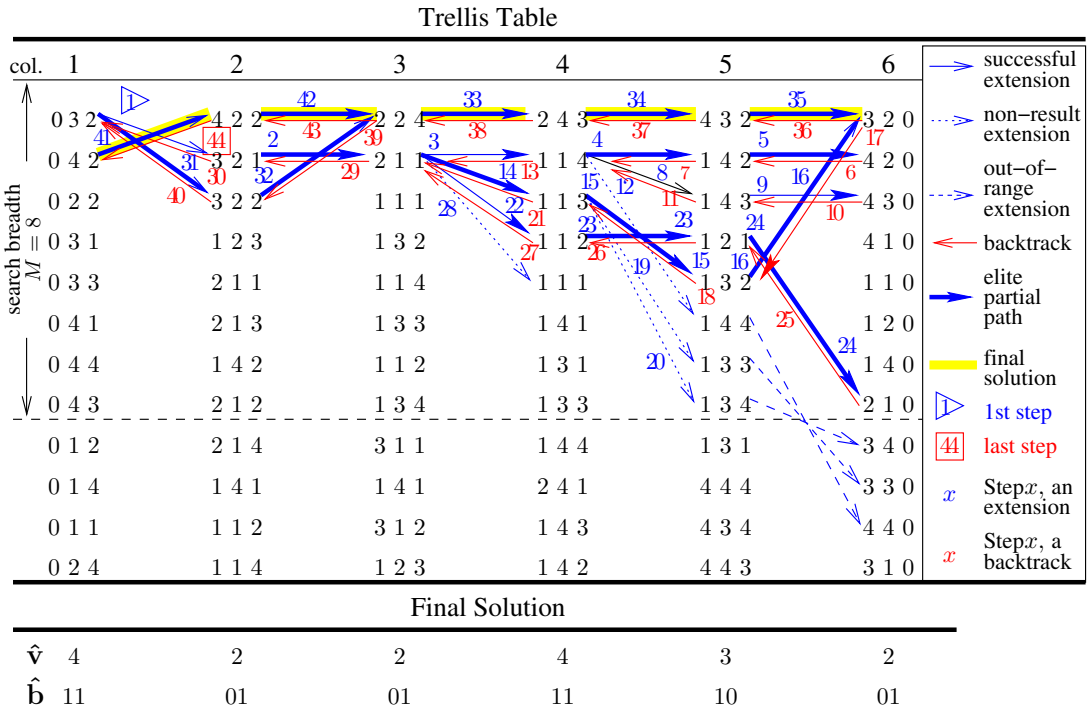

Fig. 2. An example of our decoding algorithm manipulated on a trellis table with $M = 8$ and $T = 6$. The number, regardless of whether in blue or red, indicates the sequence number of the manipulating steps. On one hand, the blue numbers label all the exploring operations. On the other hand, the red 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.

$\hat{\gamma}_1 = \arg\max_{\gamma_1 \in \{1,\ldots,N_g\}} \{\hat{\bar{q}}_{\gamma_1}^{(1)}\}$, with $N_g = 2^{2Q}$ $\langle N_g = 2^{2L} \rangle$. The above procedure is precisely expressed with pseudo codes in Alg. 1. An illustration of the whole process can be found in Fig. 2, 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. 2 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. 2.

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. 2, 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, 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. 2, 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

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

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

$i \leftarrow 0$; $p_t \leftarrow 0$
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**
      $\hat{\bar{\mathbf{v}}}_{\gamma_t}^{(t)} \leftarrow [\underbrace{0, \ \cdots, \ 0}_{T-2}, \ u_{j1}, \ \cdots, \ u_{j\frac{N_b+1}{2}}]$
      $\hat{q}_{\gamma_t}^{(t)} \leftarrow f_T(\mathbf{u}_j, \mathbf{z}_t)$
      $p_t \leftarrow 1$; **return**
    else
      $\gamma_{t+1} \xleftarrow{\text{decide}} [u_{j2}, \ \cdots, \ u_{jN_b}]$
      if $\hat{q}_{\gamma_{t+1}}^{(t+1)}$ *has NOT been calculated* **then**
      $\left\{p_{t+1}, \hat{\bar{\mathbf{v}}}_{\gamma_{t+1}}^{(t+1)}, \hat{q}_{\gamma_{t+1}}^{(t+1)}\right\} \leftarrow$
      SearchRt$(t+1, \gamma_{t+1})$
      else $p_{t+1} \leftarrow 1$
      if $p_{t+1} = 1$ **then**
        Straight Top Break
        $\bar{\mathbf{v}}_{\gamma_t,i}^{(t)} \leftarrow$
        $[\underbrace{0, \ \cdots, \ 0}_{t-2}, \ u_{j1}, \ \hat{\bar{v}}_{\gamma_{t+1}t}^{(t+1)}, \ \cdots, \ \hat{\bar{v}}_{\gamma_{t+1}T}^{(t+1)}]$
        $q_{\gamma_t,i}^{(t)} \leftarrow f_t(\mathbf{u}_j, \mathbf{z}_t) + \hat{q}_{\gamma_{t+1}}^{(t+1)}$
$k \leftarrow \arg\max_{i=1,\cdots,N_g} \left\{q_{\gamma_t,i}^{(t)}\right\}$
$\hat{q}_{\gamma_t}^{(t)} \leftarrow q_{\gamma_t,k}^{(t)}$
$\hat{\bar{\mathbf{v}}}_{\gamma_t}^{(t)} \leftarrow \bar{\mathbf{v}}_{\gamma_t,k}^{(t)}$
$p_t \leftarrow 1$; **return**

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

input : M, Δ, fitness table
output: $\hat{\mathbf{v}}$

$n \leftarrow 0$
while $n = 0$ **do**
  $M \leftarrow M + \Delta$
  for $j \leftarrow 1$ **to** M **do**
    $\gamma \xleftarrow{\text{decide}} [u_{j2}, \ \cdots, \ u_{jN_b}]$
    $\left\{p, \hat{\bar{\mathbf{v}}}_{\gamma}^{(2)}, \hat{q}_{\gamma}^{(2)}\right\} \leftarrow$ **SearchRt**$(2, \gamma)$
    if $p = 1$ **then**
      $n = n + p$
      Straight Top Break
      $\bar{\mathbf{v}}_n^{(1)} \leftarrow \hat{\bar{\mathbf{v}}}_{\gamma}^{(2)}$
      $q_n^{(1)} \leftarrow f_1(\mathbf{u}_{j_1}, \mathbf{z}_1) + \hat{q}_{\gamma}^{(2)}$
$k \leftarrow \arg\max_{i=1\cdots n} \left\{q_i^{(1)}\right\}$
$\hat{\mathbf{v}} \leftarrow \bar{\mathbf{v}}_k^{(1)}$

rationale to activate STB at the t^{th} 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 $t = t, \ldots, T$, 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. 2, 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 $t \in [1, T]$:

$$\sum_{i=1}^{t} e_i(\mathbf{u}_{j_i}, \mathbf{z}_i) \leqslant r \tag{18}$$

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.

Table I summarizes the complexity cost by calculating each part of the local OF employed by the BATSA and SDAs quantified in (16), (17) and (14), (15) respectively. First of all, the complexity recorded in Table I is quantified in real-number FLoating-point Operations Per Second (FLOPS). However, each symbol in (14)$\sim$(17) is complex-valued. Therefore, the number of times counted for complex-valued operations needs to be transferred to that of the real-number operations. Besides, the bold symbols in (14) and (16) are either vectors or matrices, and the FLOPS presented in Table I associated with any step is the superposition of the FLOPS imposed by operations manipulated by all the elements of a vector or matrix. Thirdly, (15) and (17) quantifies the local OF of only one detector, while the FLOPS in Table I is the sum of all the $\bar{T}$ independent detectors operated for a block of signals. We should point out that by defining $\mathbf{a} = \mathbf{X}\bar{\dot{\mathbf{b}}}_j + \sum_{i\in\mathbb{I}} \tilde{\mathbf{X}}_i \tilde{\dot{\mathbf{b}}}_j$ (for ACLDC) or $a = x\bar{\dot{b}}_j + \sum_{i\in\mathbb{I}} \tilde{x}_i \tilde{\dot{b}}_j(i)$ (for TR-STBC), the resultant scalar of $(\mathbf{a}^H\mathbf{a})$ or $(a^H a)$ needs to be calculated only once for each row index $j = 1, 2, \ldots, N_{\text{r}}$ for the entire $(N_{\text{r}} \times T)$ trellis table. This is because $\mathbf{a}$ or a is irrelevant with the column index t. In addition, the first part of both local OFs (16) and (17) employed by BATSA requires calculating only the real component of the complex-valued results, which halves the obliged complexity. As can be observed from the last two rows of Table I, as long as the block-length $T > 1$, the complexity imposed by calculating the BATSA local OF will always be smaller than that of the SDAs. More detailed discussion of the complexity imposed by local OFs can be found in [17].

The other differences have been scattered in the previous text, and will be summarized in Table II. 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 $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}}\in\mathbb{V}}\left\{F(\dot{\mathbf{v}})\right\}$.

Proof: See Appendix. ■

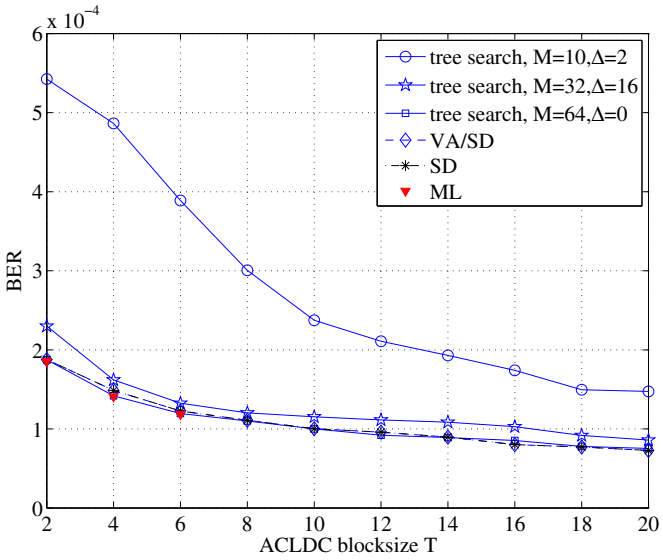

Fig. 3. BER performance versus the ACLDC block size T, when different detection algorithms are applied at system SNR= 20dB, in combination with the ACLDC parameters $K = 2$, $\bar{T} = 2$, $Q = 2$ and delay $\tau = 3/4$. The total number of ACLDC blocks used for simulation is sufficient to detect 100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

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.

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

[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 II
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

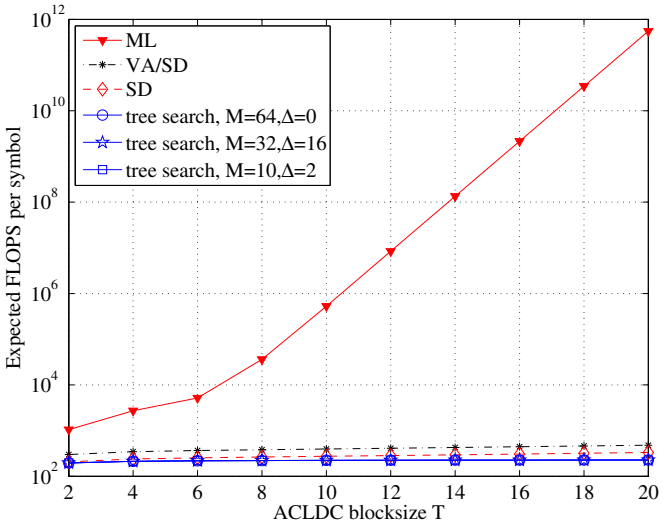

Fig. 4. Complexity versus the ACLDC block size T, when different detection algorithms are applied at system SNR= 20dB, in combination with the ACLDC parameters $K=2$, $\bar{T}=2$, $Q=2$ and delay $\tau=3/4$. The complexity of all tree-search algorithms are obtained through simulation, which is calculated 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. .

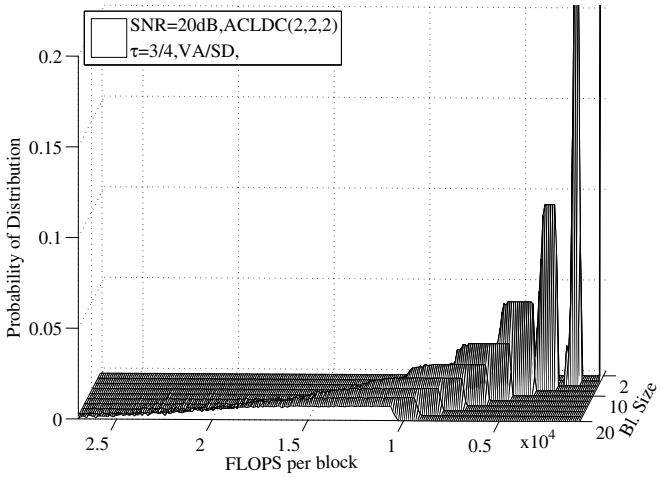

Fig. 5. Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial block size T, when VA combined SD algorithm is applied. All the point in this figure is obtained at system SNR= 20dB in conjunction with the ACLDC parameters $K=2$, $\bar{T}=2$, $Q=2$.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

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

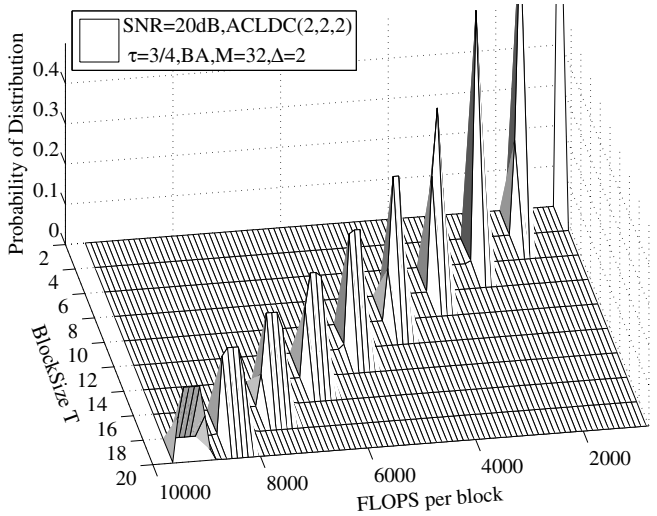

Fig. 6. Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial block size T, when our algorithm is applied with an initial searching breadth of $M = 32$ and an incremental searching step of $\Delta = 2$. All the point in this figure is obtained at 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.

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.

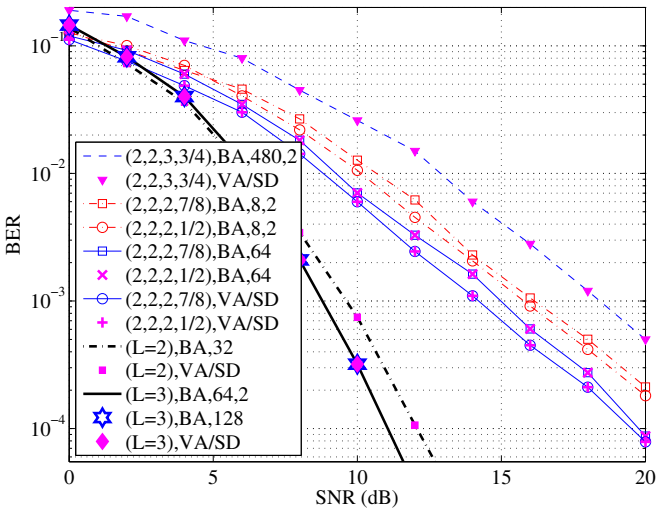

Fig. 7. BER performance versus the system SNR, when different detection algorithms are applied in systems employing an ACLDC scheme associated with $T = 20$ or an TR-STBC with $T = 8$ codewords per block, while the frequency selective channel has $L = 3$ or $L = 2$ delay paths. The numbers in the brackets 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.

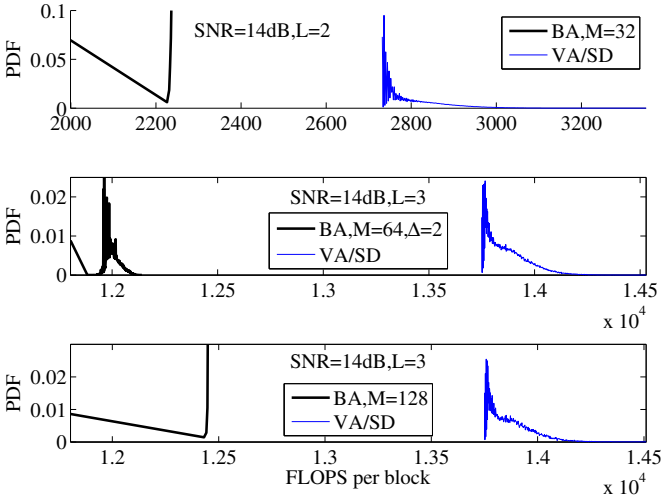

Fig. 8. Comparison of PDF of FLOPS per block of the VA/SD algorithm and our algorithm in TR-STBC assisted system obtained at SNR= 14dB, while the frequency selective channel has 3 or 4 paths and each block contains $T = 8$ codewords. A searching breadth of $M = 64$ or $M = 128$ is employed when TR-STBC 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.

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

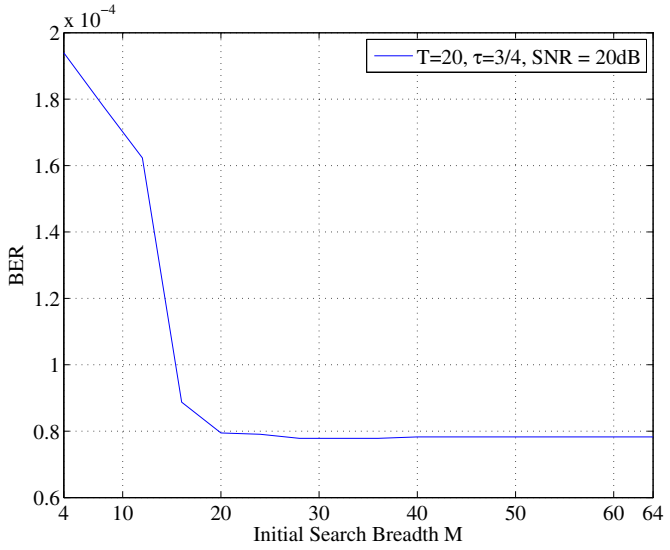

Fig. 9. BER performance versus various initial searching breadth M at a given $\Delta = 2$, when an ACLDC block is made up of $T = 20$ CLDC codewords and one guard interval as the delay of the asynchronous node is $\tau = 3/4$. Additionally, all the different detection algorithms are applied at system 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.

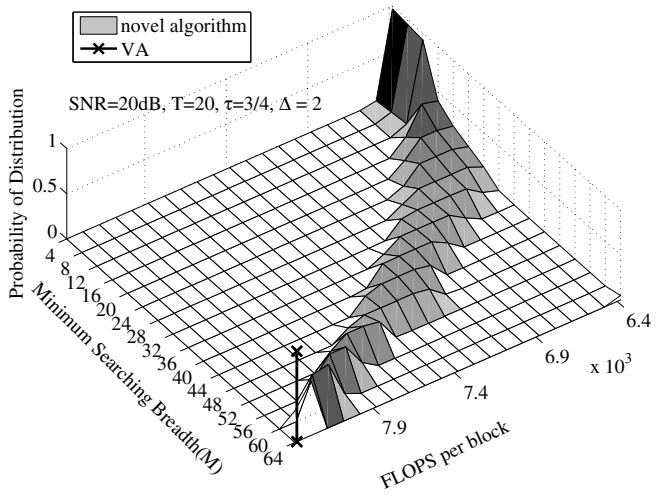

Fig. 10. Three-dimensional probability density as a function of the FLOPS cost per ACLDC block as well as the initial searching breadth M, when incremental step is $\Delta = 2$. One ACLDC block is made up of $T = 20$ CLDC codewords and one guard interval. All the point in this figure is obtained at system 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.

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.

Appendix

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:

$$\hat{\bar{\mathbf{v}}}_{\gamma}^{(\dot{t})} = \arg \max_{\dot{\mathbf{v}} \in \mathbb{V}_{\gamma}^{(\dot{t})}} \left\{ \bar{F}_{\dot{t}}(\dot{\mathbf{v}}) \right\}, \tag{19}$$

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) (19) is achieved when $\dot{t} = T - 1$;
2) If (19) is achieved when $\dot{t} = t$, and it is also satisfied when $\dot{t} = t - 1$.

Proof: We will present the proof of (19) following the structure of mathematical induction. We will firstly clarify some notations. $N_{\mathrm{u}} = 2^Q$ (for ACLDC) or $N_{\mathrm{u}} = 2$ (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. $N_{\mathrm{b}} = 3$ (for ACLDC) or $N_{\mathrm{b}} = 2L+1$ (for TR-STBC) denotes the number of elements contained by the vector representing each node in the trellis table. $N_{\mathrm{g}} = 2^{2Q}$ (for ACLDC) or $N_{\mathrm{g}} = 2^{L+1}$ (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^{th} column. Hence, by assuming $k = \arg\max_{i=1,\ldots,N_{\mathrm{u}}} \{ f_{T-1}(\mathbf{u}_{\gamma_{(T-1)},i}) + f_T(\mathbf{u}_{\gamma_T}) \}$, where γ_T is decided by the last $N_{\mathrm{b}} - 1$ elements of $\mathbf{u}_{\gamma_{(T-1)},i}$, we have $\hat{\bar{\mathbf{v}}}_{\gamma_{(T-1)}}^{(T-1)} = \bar{\mathbf{v}}_{\gamma_{(T-1)},k}^{(T-1)} = \arg\max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}_{\gamma_{T-1}}^{(T-1)}} \left\{ \bar{F}_{T-1}(\dot{\mathbf{v}}) \right\}$. That is $\hat{\bar{\mathbf{v}}}_{\gamma}^{(t)} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}_{\gamma}^{(t)}} \left\{ \bar{F}_t(\dot{\mathbf{v}}) \right\}$ is achieved when $t = T - 1$.
2) When $\dot{t} = t$, there will be N_{u} nodes $\mathbf{u}_{\gamma_{t-1},i} \in \mathbb{U}_{\gamma_{t-1}}$ with $i = 1, \ldots, N_{\mathrm{u}}$ in the $(t-1)^{\text{th}}$ column. According to the algorithm description

$$q_{\gamma_{t-1},i}^{(t-1)} = f_{t-1}(\mathbf{u}_{\gamma_{t-1},i}) + \hat{q}_{\gamma_{t,i}}^{(t)}, \quad \forall i = 1, \ldots, N_{\mathrm{u}} \tag{20}$$

where $\gamma_{t,i}$ is determined by the last $(N_\mathrm{b}-1)$ bits of $\mathbf{u}_{\gamma_{t-1},i}$, and we have $\gamma_{t,1} \neq \gamma_{t,2} \neq \cdots \neq \gamma_{t,N_\mathrm{u}}$. Since (19) is maintained when $\dot{t}=t$, we should have

$$\hat{\dot{\mathbf{v}}}^{(t)}_{\gamma_{t,i}} = \arg \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(t)}_{\gamma_{t,i}}} \left\{ \bar{F}_t(\dot{\mathbf{v}}) \right\}, \forall i = 1, \cdots, N_\mathrm{u} \quad (21)$$

As given by the definition of $\hat{q}^{(t)}_{\gamma}$ and (21), we should have $\hat{q}^{(t)}_{\gamma_{t,i}} = \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(t)}_{\gamma_{t,i}}} \left\{ \bar{F}_t(\dot{\mathbf{v}}) \right\}$, $\forall, i = 1, \ldots, N_\mathrm{u}$. 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_b elements, i.e. from the $(t-1-(N_\mathrm{b}-1)/2)^{\text{th}}$ to the $(t-1+(N_\mathrm{b}-1)/2)^{\text{th}}$ elements equaling to $\mathbf{u}_{\gamma_{t-1},i}$. Therefore, the regional elite partial vector $\hat{\dot{\mathbf{v}}}^{(t-1)}_{\gamma_{t-1},i}$ having the highest accumulated fitness value in $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$ should be $\hat{\dot{\mathbf{v}}}^{(t)}_{\gamma_{t,i}}$, with its $(t-1-(N_\mathrm{b}-1)/2)^{\text{th}}$ element changed to $u_{\gamma_{t-1},i,1}$, yielding $\bar{\mathbf{v}}^{(t)}_{\gamma_{t,i}}$. That is: $\bar{\mathbf{v}}^{(t-1)}_{\gamma_{t-1},i} = \arg\max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}} \{\bar{F}_{t-1}(\dot{\mathbf{v}})\}, \forall i = 1, \ldots, N_\mathrm{u}$. According to the definition of $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$, the $(t-1-(N_\mathrm{b}-1)/2)^{\text{th}}$, $\cdots$, $(t-1+(N_\mathrm{b}-1)/2)^{\text{th}}$ 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 (20) we have

$$q^{(t-1)}_{\gamma_{t-1},i} = \max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}} \{\bar{F}_{t-1}(\dot{\mathbf{v}})\}, \forall i = 1, \cdots, N_\mathrm{u}. \quad (22)$$

According to (22) and the definition of $\hat{q}^{(t-1)}_{\gamma_{t-1}}$, we have

$$\begin{aligned} \hat{q}^{(t-1)}_{\gamma_{t-1}} &= \max_{i=1,\cdots,N_\mathrm{u}} \left\{ q^{(t-1)}_{\gamma_{t-1},i} \right\} \\ &= \max_{i=1,\cdots,N_\mathrm{u}} \left\{ \max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}} \{\bar{F}_{t-1}(\dot{\mathbf{v}})\} \right\}. \end{aligned} \quad (23)$$

Since $\bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1}} = \bigcup_{i=1}^{N_\mathrm{u}} \bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1},i}$, (23) can be further simplified as: $\hat{q}^{(t-1)}_{\gamma_{t-1}} = \max_{\dot{\mathbf{v}} \in \bar{\mathbb{V}}^{(t-1)}_{\gamma_{t-1}}} \{\bar{F}_{t-1}(\dot{\mathbf{v}})\}$. Therefore, $\hat{\dot{\mathbf{v}}}^{(t-1)}_{\gamma_{t-1}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}^{(t-1)}_{\gamma_{t-1}}} \left\{ \bar{F}_{t-1}(\dot{\mathbf{v}}) \right\}$. So far, we have proved that as long as (19) 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, (19) will always be true within the range of $\dot{t}=2,\ldots,T-1$. We will further prove $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}} \left\{ F(\dot{\mathbf{v}}) \right\}$. For any given node $\mathbf{u}_\gamma$ in the first column, we define the set entailing vectors starting with $\mathbf{u}_\gamma$ as set $\mathbb{V}^{(1)}_\gamma$. Apparently, we have $\mathbb{V}^{(1)}_\gamma = \bar{\mathbb{V}}^{(2)}_\gamma$. As further derived from (19), we have

$$\hat{\dot{\mathbf{v}}}^{(2)}_{\gamma} = \arg \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(2)}_{\gamma}} \left\{ \bar{F}_2(\dot{\mathbf{v}}) \right\}, \forall \gamma = 1 \cdots N_\mathrm{g}. \quad (24)$$

Since all vectors in $\mathbb{V}^{(1)}_\gamma$ have the same first $(N_\mathrm{b}+1)/2$ elements, $f_1(\dot{\mathbf{v}})$ will be the same for any vector $\dot{\mathbf{v}} \in \mathbb{V}^{(1)}_\gamma$. Therefore, as suggested by Line 11 of Alg. 2 and (24),

$$\begin{aligned} q^{(1)}_\gamma &= \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(1)}_\gamma} \left\{ f_1(\dot{\mathbf{v}}) + \bar{F}_2(\dot{\mathbf{v}}) \right\} = \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(1)}_\gamma} \left\{ \bar{F}_1(\dot{\mathbf{v}}) \right\} \\ &= \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(1)}_\gamma} \left\{ F(\dot{\mathbf{v}}) \right\}, \forall \gamma = 1 \cdots N_\mathrm{g} \end{aligned} \quad (25)$$

According to Lines 12 and 13 in Alg. 2 and (25), we will have

$$\hat{\mathbf{v}} = \arg \max_{\gamma=1\cdots N_\mathrm{g}} \left\{ q^{(1)}_\gamma \right\} = \arg \max_{\gamma=1\cdots N_\mathrm{g}} \left\{ \max_{\dot{\mathbf{v}} \in \mathbb{V}^{(1)}_\gamma} \left\{ F(\dot{\mathbf{v}}) \right\} \right\}. \quad (26)$$

We have $\mathbb{V} = \bigcup_{\gamma=1}^{N_\mathrm{g}} \mathbb{V}^{(1)}_\gamma$ and (26) can be further simplified as: $\hat{\mathbf{v}} = \arg\max_{\dot{\mathbf{v}} \in \mathbb{V}} \left\{ F(\dot{\mathbf{v}}) \right\}$. 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. ■

REFERENCES

[1] E. Lindskog and A. Paulraj, "A transmit diversity scheme for channels with intersymbol interference," in *Proc. 2000 IEEE International Conf. Commun.*, 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 *Proc. 2009 IEEE International Conf. Inf., Commun. Signal Process.*, 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. Lett.*, 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 Trans. Inf. 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 Trans. Veh. Technol.*, vol. 59, no. 2, pp. 779–787, Feb. 2010.

[6] M. O. Damen and A. R. Hammons, "Delay-tolerant distributed-TAST codes for cooperative diversity," *IEEE Trans. Inf. 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 Trans. Commun.*, vol. 32, no. 2, pp. 169–176, Feb. 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 Trans. Wireless Commun.*, vol. 4, pp. 710–721, May 2005.

[9] Y. Kim and K. Cheun, "A reduced-complexity tree-search detection algorithm for MIMO systems," *IEEE Trans. Signal Process.*, 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 J. Sel. Topics Signal Process.*, pp. 1438–1450, Dec. 2011.

[11] K.-C. Lai and L.-W. Lin, "Low-complexity adaptive tree search algorithm for MIMO detection," *IEEE Trans. Wireless Commun.*, 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 J. 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 Trans. Commun.*, vol. 54, no. 7, pp. 1179–1183, July 2006.

[14] H. Vikalo and B. Hassibi, "Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding," *EURASIP J. Appl. Signal Process.*, 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 Trans. Signal Process.*, 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 Trans. Signal Process.*, vol. 38, no. 3, Mar. 2010.

[17] C. Xu and H. Gharavi, "A comprehensive comparison between a novel tree-search algorithm and the sphere-decoding based algorithms," Technique Report. Available: http://w3.antd.nist.gov/wctg/adhocvideo/html/tree-search-report.pdf.

[18] B. Hassibi and B. M. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1804–824, July 2002.

Chong Xu (M'10) received the B.Eng. degree in communication engineering from Beijing University of Posts and Telecommunication, Beijing, China, in 2004 and the M.Sc. degree in Radio Frequency communication systems and the Ph.D. degree in wireless communications from the University of Southampton, Southampton, U.K., in 2005 and 2009, respectively. Since March 2010, she has been with the National Institute of Standards and Technology (NIST), US Department of Commerce, Gaithersburg, MD. Her research interests include multiuser detection, cooperative transmission, asynchronous receiver, MIMO, CDMA, multi-carrier CDMA, Turbo detection, non-linear optimization algorithms. Dr. Xu has been involved in the Smart Grid, Scheduling/Routing algorithms-design projects in NIST. She received the Dorothy Hodgkin Postgraduate Awards (DHPA) in 2005 in the U.K.

Hamid Gharavi (F'92) received the Ph.D. degree from Loughborough University, Loughborough, U.K., in 1980. He joined AT&T Bell Laboratories, Holmdel, in 1982. He was then transferred to Bell Communications Research (Bellcore) after the AT&T-Bell divestiture, where he became a Consultant on video technology and a Distinguished Member of Research Staff. In 1993, he joined Loughborough University as Professor and Chair of Communication Engineering. Since September 1998, he has been with the National Institute of Standards and Technology (NIST), US Department of Commerce, Gaithersburg, MD. Dr. Gharavi was a core member of the Study Group XV (Specialist Group on Coding for Visual Telephony) of the International Communications Standardization Body CCITT (ITU-T). He was selected as one of the six university academics to be appointed to the U.K. Government's Technology Foresight Panel in Communications to consider the future through 2015 and make recommendations for allocation of key research funds. His research interests include smart grid, wireless multimedia, mobile communications and wireless systems, mobile ad-hoc networks, and video/image transmission. He holds eight U.S. patents related to these topics. Dr Gharavi received the Charles Babbage Premium Award from the Institute of Electronics and Radio Engineering in 1986, and the IEEE CAS Society Darlington Best Paper Award in 1989. He has been a Distinguished Lecturer of the IEEE Communication Society. In 1992 Dr. Gharavi was elected a Fellow of IEEE for his contributions to low bit-rate video coding and research in subband coding for image and video applications. He has been a Guest Editor for a number of special issues, including one on Smart Grid, which has been published in June, 2011 by the PROCEEDINGS OF THE IEEE. He served as a member of the Editorial Board of the PROCEEDINGS OF THE IEEE from January, 2003 to December, 2008. He is currently a member of the Editorial board, *IET Image Processing*. Dr Gharavi served as an Associate Editor for the IEEE TRANSACTIONS ON CAS FOR VIDEO TECHNOLOGY (CSVT) from 1996 to 2006. He then became the Deputy Editor-in-Chief of this IEEE Transactions through December 31, 2009. Currently, he is serving as the Editor-in-Chief for the IEEE TRANSACTIONS ON CSVT.