J Sign Process Syst (2011) 62:427–442
DOI 10.1007/s11265-010-0474-9

Efficient Detection Algorithms for MIMO Communication Systems

Di-You Wu · Lan-Da Van

Received: 17 June 2009 / Revised: 4 March 2010 / Accepted: 8 March 2010 / Published online: 4 April 2010

Abstract In this paper, two new efficient detection algorithms, Type 1 (T1) with better complexity-performance tradeoff and Type 2 (T2) with lower complexity, are derived from one generalized framework for multiple-input multiple-output (MIMO) communication systems. The proposed generalized detection framework constructed by parallel interference cancellation (PIC), group, and iteration techniques provides three parameters and three sub-algorithms to generate two efficient detection algorithms and conventional BLAST-ordered decision feedback (BODF), grouped, iterative, and B-Chase detection algorithms. Since the group interference suppression (GIS) technique is applied to the proposed detection algorithms, the complexities of the preprocessing (PP) and tree search (TS) can be reduced. In (8,8) system with uncoded 16-QAM inputs, one example of the T1 algorithm can save complexity by 21.2% at the penalty of 0.6 dB loss compared with the B-Chase detector. The T2 algorithm not only reduces complexity by 21.9% but also outperforms the BODF algorithm by 3.1 dB.

Keywords Chase detection · Group detection · Group interference suppression · Iterative detection · Multiple-input multiple-output (MIMO) · Sorted-QR decomposition · Vertical Bell Laboratories layered space-time (V-BLAST)

Preliminary results were presented in Proceedings of *IEEE Vehicular Technology Conference (VTC)*, Calgary, Canada, Sep. 2008.

D.-Y. Wu (✉) · L.-D. Van
Department of Computer Science,
National Chiao Tung University,
Hsinchu, Taiwan 300, Republic of China
e-mail: dywu@viplab.cs.nctu.edu.tw

L.-D. Van
e-mail: ldvan@cs.nctu.edu.tw

1 Introduction

Multiple-input multiple-output (MIMO) technology can significantly improve data transmission rate in bandwidth-limited wireless communications without increasing the transmission power. Much research [1, 2] has shown that the channel capacity increases while the number of antennas is raised. Because of the above advantage, the MIMO technique has been considered in modern high-speed wireless communication standard including wireless LAN [3] and mobile wireless MAN. The Bell Laboratories layered space-time (BLAST) wireless communication system [1] uses multi-element antenna arrays at both the transmitter and receiver to achieve high spectral efficiency. This technology is referred to as the diagonal BLAST (D-BLAST). The D-BLAST theoretically approaches the Shannon capacity for multiple transmitters and receivers, but the D-BLAST is complex and impractical in hardware design. The vertical BLAST (V-BLAST) system [4, 5] is a simplified architecture of the D-BLAST, where the BLAST-ordered decision feedback (BODF) detection algorithm named in [6] (also called successive interference cancellation (SIC) detection algorithm named in [7]) is applied. Although the BODF algorithm [8, 9] has low computational complexity, the bit-error rate (BER) performance is not satisfactory. In terms of BER performance in the MIMO detection system, the maximum likelihood (ML) detection scheme is an optimum solution to the receiver. However, it is manifest that the detection complexity significantly raises as the number of antennas and the constellation size increase. Thus, the ML scheme is not suitable for high-speed hardware implementation. The sphere decoding (SD) scheme [10, 11] searching for the closest lattice point inside the bounded radius of sphere achieves the same ML detection performance with efficient computational complexity. An efficient

SD algorithm [12] improved the computational complexity by the projection scheme with sacrificing BER performance. However, the complexity of the SD algorithm is unstable owing to the variation of the iteration number which is sensitive to the signal-to-noise ratio (SNR). Especially, while the SD algorithm requires larger iteration number, the higher computational complexity at low SNR environment is incurred. Thus, the variable throughput of the SD algorithm affects the system performance.

Currently, there have been many studies [6, 13–23] on developing detection algorithms in both complexity and performance between the ML and BODF detection algorithms. The research work [13, 14] divides symbols into two groups. After group partition, the first-group symbols are detected by the ML detection and the second-group symbols are detected by a suboptimal algorithm after cancelling the interference from the first-group symbols. Although the previously published schemes [13, 14] using the ML and suboptimal detection algorithms can achieve better performance, the high computational complexity is incurred. The B-Chase detection algorithm [6] that lists more candidates for the parallel detection shows good performance and complexity trade-off for different demands. However, while increasing the number of antennas, the preprocessing (PP) complexity of the B-Chase algorithm largely increases because a full size channel matrix is processed without matrix decomposition. On the other hand, under the limited complexity, the B-Chase algorithm, in most cases, cannot choose better candidates from adjacent points for detection. Due to above reasons, we are motivated to propose two efficient detection algorithms with the group interference suppression (GIS) technique via one generalized detection framework constructed by the parallel interference cancellation (PIC), group, and iteration schemes. Compared with the B-Chase algorithm, the proposed Type 1 (T1) algorithm features the low computational complexity and satisfactory performance for large number of antennas. The proposed Type 2 (T2) algorithm shows lower complexity compared with the BODF and T1 algorithms.

This paper is organized as follows. Brief review of the MIMO detection algorithms is described in Section 2. In Section 3, one generalized detection framework has been presented. In the same section, how to generate existing detection algorithms through this framework will be discussed. In Section 4, we propose two new efficient detection algorithms via the generalized framework. The parallel characteristic comparison, complexity analysis and performance simulation results are presented in Section 5. Finally, the conclusions are remarked in the last section. Several acronyms used in this paper are listed in Appendix.

2 Brief Review of the MIMO Detection Algorithms

An (N,M) MIMO system with N transmit antennas and M receive antennas is considered in this paper. The discrete-time received signal $\mathbf{r}$ can be written as

$$\mathbf{r} = \mathbf{Hs} + \mathbf{n}, \tag{1}$$

where $\mathbf{s}$ denotes the $N\times 1$ vector of the simultaneous transmitted symbols that select from constellation C, and $|C|$ denotes the constellation size. $\mathbf{H}$ is the $M\times N$ channel matrix, and $\mathbf{n}$ is the $M\times 1$ complex noise vector. In this paper, the elements in $\mathbf{H}$ are assumed to be independent identically distributed (IID) complex Gaussian random variable with zero mean, where the dimension is under $M\geq N$. It is assumed that the receiver knows channel matrix $\mathbf{H}$ perfectly and is known that the ML detector is an optimum solution to the receiver. The ML scheme detects all sub-stream symbols jointly by choosing the symbol vector that maximizes likelihood function. The above treatment is equivalent to the minimum Euclidean distance (MED) function in (2).

$$\mathbf{s} = \arg\min_{i} \|\mathbf{r} - \mathbf{Hs}_i\|^2, \tag{2}$$

where $||\mathbf{x}||$ denotes 2-norm of the vector $\mathbf{x}$ and $\mathbf{s}_i$ denotes the i-th candidate choosing from all possible combinations of symbols. Note that the number of all combinations is $|C|^N$. Nevertheless, the ML scheme with high computation-complexity blocks the VLSI implementation. Many low-complexity detection algorithms [13–23] have been widely studied. Herein, we briefly review the complexity-oriented algorithms as follows.

2.1 Grouped Detection (GD)

The grouped detection algorithm [14] applies the ordering, GIS [15], ML algorithm to the first group symbols, interference canceling (IC), and BODF algorithm to the second group symbols. The GIS not only plays the role of dividing symbols into two groups but also suppresses the performance influence of the low SNR signals. After ordering symbols, the ML detection algorithm is employed to detect higher SNR signals for the first group. Because of the property of the ML algorithm, we can detect symbols at the early stage and guarantee the performance without error propagation. The remaining symbols of the second group disturbed by high noise power are detected by a suboptimal algorithm such as the BODF detection algorithm [4, 5, 8, 9]. A multi-group detection algorithm [16] has been proposed to enhance the BER performance by increasing diversity for each group; however, the computational complexity increases due to multiple-GIS computations.

2.2 Iterative Detection (ID)

Since the traditional BODF algorithm propagates errors to next detected symbol through the operation of interference cancellation if the previous detected symbols are in errors, the overall system performance is confined. In order to decrease the error propagation, the iterative detection algorithm [17, 18] was proposed to enhance diversity for all symbols. The iterative detection algorithm detects symbols repeatedly in a specific sequence such that low-diversity symbols can be redetected to obtain the high diversity gain.

2.3 Chase Detection

The Chase detection framework [6, 19] determines four parameters: which symbol detected first, list length, filter type, and sub-detector algorithm for the MIMO detection. Many detection algorithms including ML, BODF, parallel [20], B-Chase and S-Chase can be derived from the Chase detection framework by adjusting above four parameters. The B-Chase detection based on the BODF algorithm provides a tradeoff between the complexity and performance by choosing the list length. When the list length equals the constellation size, the performance of the B-Chase detection is close to that of the ML detection.

2.4 GPIC Detection

The generalized parallel interference cancellation (GPIC) detection algorithm uses two parallel interference cancellation (PIC) techniques; one is the same as that of the B-Chase detection algorithm and another is referred to as a redetection scheme. For the first PIC technique, the GPIC extends the number of first detected symbols compared with the B-Chase detection. In this case, the number of list lengths is the same as the number of all possible combinations of the first detected symbols. For the second PIC technique, the GPIC detection applies the redetection scheme to detect residual symbols again, where the redetection scheme uses linear detection (LD) algorithm for lower computational complexity.

3 Generalized MIMO Detection Framework

In terms of the BER performance between GD and ID algorithms, we simulate the comparison results as shown in Fig. 1. The GD(K) denotes the GD algorithm with K symbols of the first group, and the ID($I_{\max}$) denotes the ID algorithm with maximum iteration $I_{\max}$. Specifically, ID(1) means that the BODF algorithm [4, 5, 8, 9] is performed two times, where the second time is operated recursively. In Fig. 1, the GD(2) algorithm outperforms the ID(1) algorithm at high SNR environment, where SNR is defined as the signal power over the noise power. At low SNR environment, the GD(2) algorithm has weaker performance than the ID(1) does. On the other hand, according to the previous work mentioned in Section 2, the PIC technique of the GPIC algorithm is used to look for better solution to enhance BER performance. In order to take advantages of three algorithms and attain the low complexity with satisfactory performance, the proposed generalized framework adopts PIC, group, and iteration techniques. In this framework, the symbols are partitioned into two groups: Group-I and Group-II. To easily understand the framework, three parameters and three sub-algorithms are defined first as follows.

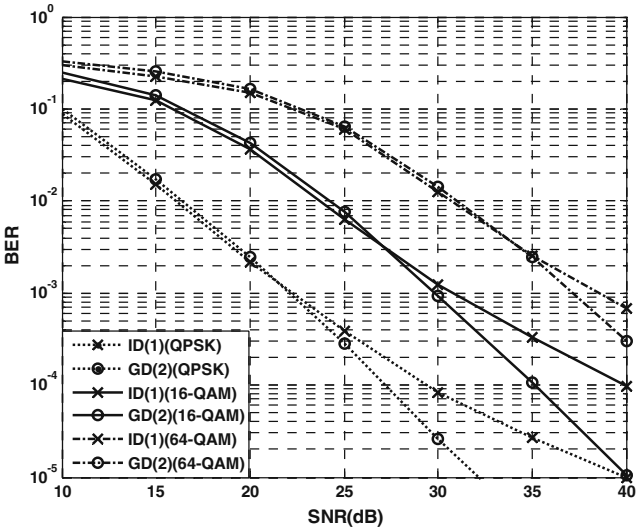

Figure 1 BER performance comparison with GD and ID algorithms in (8,8) MIMO system.

- K: Number of symbols in Group-I whose range is $1 \le K < N$.
- ℓ: List length whose value is $1 \le \ell \le |C|^K$.
- $I_{\max}$: Maximum number of iterations whose number is $I_{\max} \ge 0$.
- sa_1, sa_2, and sa_3: Sub-detection algorithms used in the generalized framework.

The proposed generalized framework as shown in Fig. 2 consists of six steps. Each step is illustrated in the following.

Step 1: Order and partition all symbols into two groups. Group-I has K symbols $\{s_{n_1}, s_{n_2}, \cdots, s_{n_K}\}$, and the other ($N$-$K$) symbols $\{s_{n_{K+1}}, s_{n_{K+2}}, \cdots, s_{n_N}\}$ belong to Group-II.

Step 2: Determine a list of partial candidates $\left\{\mathbf{s}'_{I_1}, \mathbf{s}'_{I_2}, \cdots, \mathbf{s}'_{I_\ell}\right\}$ for the Group-I symbols by sa_1, where $\mathbf{s}'_{I_i} = \left[s'_{i,n_1} s'_{i,n_2} \cdots s'_{i,n_K}\right]^T$, and $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$.

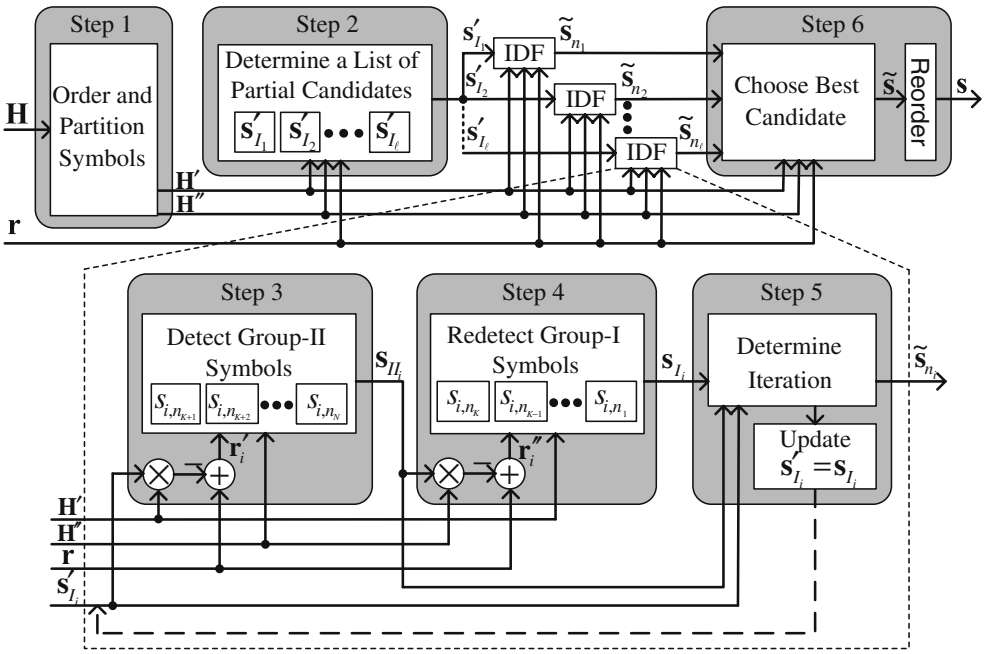

Figure 2 Block diagram of the generalized framework.

Step 3: Cancel the interference of K symbols for each $\mathbf{s}_{I_i}^{'}$ from $\mathbf{r}$ to derive $\mathbf{r}_i^{'}$, and detect the remaining $(N\text{-}K)$ symbols $\mathbf{s}_{II_i} = \left[s_{i,n_{K+1}} s_{i,n_{K+2}} \cdots s_{i,n_N}\right]^T$ by sa_2.

Step 4: Cancel the interference of $(N\text{-}K)$ symbols for each $\mathbf{s}_{II_i}$ from $\mathbf{r}$ to derive $\mathbf{r}_i^{''}$, and redetect the K symbols $\mathbf{s}_{I_i} = \left[s_{i,n_1} s_{i,n_2} \cdots s_{i,n_K}\right]^T$ by sa_3.

Step 5: Determine whether the iterative operation is activated. Once the iteration is triggered, the framework will update the parameter values and go back to Step 3. When there is no iteration, we combine $\mathbf{s}_{I_i}$ and $\mathbf{s}_{II_i}$ into the i-th candidate $\widetilde{\mathbf{s}}_{n_i}$.

Step 6: Choose the best hard decision $\widetilde{\mathbf{s}}$ among the candidates $\{\widetilde{\mathbf{s}}_{n_1}, \widetilde{\mathbf{s}}_{n_2}, \cdots, \widetilde{\mathbf{s}}_{n_\ell}\}$ by the MED criterion in (2), and then reorder $\widetilde{\mathbf{s}}$ into $\mathbf{s}$.

We treat Steps 3∼5 as an iterative decision feedback (IDF) block that detects two group symbols, repeatedly. Since more candidates are generated in Step 2, IDFs can be operated in parallel. The trigger condition in Step 5 depends on different detection algorithms. For example, in the case of ID algorithm, if $s_{i,n_1}^{'}$ is not equal to s_{i,n_1} and iteration number does not reach to $I_{\max}$, the iteration will be triggered. In this framework, PIC and iteration can be disabled (i.e., ℓ=1 and $I_{\max}$=0, respectively) to generate more different detection algorithms. As shown in Table 1, while $(K,\ \ell,\ I_{\max})=(1\le K<N,\ 1,\ 0)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$ (BODF, BODF, Identity), the framework can generate the BODF algorithm in [4, 5, 8, 9], where identity means that we bypass the operations at this stage and feed the symbols directly to the next step. When identity is used in Step 4, we assign $\left\{s_{i,n_1}, s_{i,n_2}, \cdots, s_{i,n_K}\right\} = \left\{s_{i,n_1}^{'}, s_{i,n_2}^{'}, \cdots, s_{i,n_K}^{'}\right\}$. Since sa_1 and sa_2 apply the same BODF algorithm, the number of symbols, K, of the Group-I can vary from 1 to N-1. While $(K,\ \ell,\ I_{\max})=(1<K<N,\ 1,\ 0)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$(ML(ZF-GIS), BODF, Identity), the framework can reduce to the GD algorithm in [14]. While $(K,\ \ell,\ I_{\max})=(N\text{-}1,\ 1,\ I_{\max}\ge 1)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$(BODF, BODF, BODF), the framework can generate the ID algorithm in [18]. While $(K,\ \ell,\ I_{\max})=(1\le K<N,\ 1\le\ell<|C|^K,\ 0)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$(LD, BODF, Identity), the framework can reduce to the list algorithm in [21]. While $(K,\ \ell,\ I_{\max})=(1,\ 1\le\ell<|C|,\ 0)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$ (LD, BODF, Identity), the framework can generate the B-Chase algorithm in [6]. While $(K,\ \ell,\ I_{\max})=(1\le K<N\ ,\ |C|^K,\ 0)$ and $(\mathrm{sa}_1,\ \mathrm{sa}_2,\ \mathrm{sa}_3)=$(ML, LD, Identity), the framework can reduce to the GPIC(K,0) algorithm in [7]. Hence, this framework can cover many conventional detection algorithms. Furthermore, two new efficient detection algorithms listed in the last two rows of Table 1 will be illustrated in next section.

4 Efficient Detection Algorithms

In this section, we explore the above framework by configuring three parameters including K, ℓ, $I_{\max}$ and three sub-algorithms including sa_1, sa_2, sa_3 and then propose two new efficient detection algorithms. From previous literature [6, 7, 20, 21], the BER performance can be improved by increasing the number of candidates of Group-I. However, the computational complexities of the Parallel algorithm [20] and GPIC algorithm [7] become heavy because all possible combinations of Group-I have to be evaluated. On the other hand, under the limited computational complexity,

Table 1 Cases of the generalized framework for MIMO detection.

Detector	Number of symbols in Group-I: K	Sub-algorithm used in Step 2: sa_1	List Length ℓ	Sub-algorithm used in Step 3: sa_2	Sub-algorithm used in Step 4: sa_3	Iteration Determination in Step 5 (I_{max})
BODF [4]	$1\le K<N$	BODF	1	BODF	Identity	No(I_{max}=0)
GD [14]	$1<K<N$	ML(ZF-GIS)	1	BODF	Identity	No(I_{max}=0)
ID [18]	$K=(N-1)$	BODF	1	BODF	BODF	Yes($I_{max}\ge1$)[a]
List [21]	$1\le K<N$	LD	$1\le\ell<\lvert C\rvert^K$	BODF	Identity	No(I_{max}=0)
		ML	$\lvert C\rvert^K$			
B-Chase [6]	K=1	LD	$1\le\ell<\lvert C\rvert$	BODF	Identity	No(I_{max}=0)
		ML	$\lvert C\rvert$			
GPIC(K,0) [7]	$1\le K<N$	ML	$\lvert C\rvert^K$	LD	Identity	No(I_{max}=0)
Proposed T1	$1<K<N$	B-Chase(Modified ZF-GIS)	$1\le\ell\le\lvert C\rvert$	SQRDF	SQRDF/Identity	Yes($I_{max}\ge1$) [b] or No(I_{max}=0)
Proposed T2	K=2	V-ML(Modified ZF-GIS)	1	SQRDF	Identity	No(I_{max}=0)

[a] If $(s'_{i,n_1} = = s_{i,n_1}$ or Iteration Number (I)==I_{max}), end; else, set $s'_{i,n_1} = s_{i,n_1}$ and iterate.
[b] If $\left(\left\{s'_{i,n_1}, s'_{i,n_2}, \cdots, s'_{i,n_k}\right\} == \left\{s_{i,n_1}, s_{i,n_2}, \cdots, s_{i,n_k}\right\}$ or $I == I_{max}\right)$,, end; else, set $\left\{s'_{i,n_1}, s'_{i,n_2}, \cdots, s'_{i,n_k}\right\} = \left\{s_{i,n_1}, s_{i,n_2}, \cdots, s_{i,n_k}\right\}$ and iterate.

the B-Chase algorithm [6] and list algorithm [21] probably cannot choose better candidates from adjacent points in most cases. In order to trade off the complexity and performance, we apply the GIS technique one time to the parallel detection such that candidates with smaller Euclidean distance (ED) can be obtained. According to this scheme, the proposed T1 and T2 algorithms can obtain better candidates of Group-I with higher probability for detection. The detailed manipulations of the T1 and T2 algorithms are described in the following.

4.1 Type 1 (T1) Detection Algorithm

In the proposed T1 detection algorithm [22], the B-Chase sub-algorithm is used as sa_1 in Step 2, where the BER performance of the B-Chase algorithm is close to that of the ML algorithm. Waters *et al.* [6] have shown that the computational complexity of the B-Chase algorithm is much lower than that of the ML algorithm. The sa_2 and sa_3 detect the Group-II and Group-I symbols in (M,N-K) and (M,K) MIMO sub-systems, respectively. When the number of outputs is larger than that of inputs of the above sub-systems, the probability of getting well-conditioned channel matrixes is higher; thus, we can obtain BER results closer to ML with higher probability. The sorted QR decision feedback (SQRDF) algorithm [24] that decides the detected order in the QR decomposition has lower complexity than the BODF algorithm that decides the detected symbol after detecting a symbol each time. Considering low computational complexity, the SQRDF algorithm [24] is regarded as sa_2 and sa_3 in the proposed T1 detection algorithm. Next, a wide range of parameters K, ℓ, and I_{max} is used to trade off the complexity and performance. The detailed implementation of design steps of the T1 detection algorithm is summarized in Figs. 3, 4, and 5. Each corresponding design step is described in the following.

Step 1: At the first step, we select K symbols with higher SNR to detect first by near-optimal algorithm such that error propagation can be alleviated. The columns of channel matrix sorted by 2-norm are expressed as

$$p_i = \left\|\mathbf{h}_{:,i}\right\|^2 \text{ for } i = 1, 2, \ldots, N, \tag{3}$$

where $\mathbf{h}_{:,i}$ is the i-th column of $\mathbf{H}$. According to the value of each p_i, we can sort the values to $p_{n_1} \ge p_{n_2} \ge \ldots \ge p_{n_N}$, where $\{n_1, n_2, \ldots, n_N\}$ denotes the detection order index. After permuting all symbols $\mathbf{s}$, channel matrix $\mathbf{H}$, and identity matrix I_N, we can recast the system function as follows

$$\mathbf{r} = \widetilde{\mathbf{H}}\widetilde{\mathbf{s}} + \mathbf{n}, \tag{4}$$

where $\widetilde{\mathbf{H}} = \mathbf{H}\mathbf{\Pi} = [\mathbf{h}_{n_1}\mathbf{h}_{n_2}\cdots\mathbf{h}_{n_N}]$, $\widetilde{\mathbf{s}} = \mathbf{\Pi}^T\mathbf{s} = [s_{n_1}s_{n_2}\cdots s_{n_N}]^T$, and $\mathbf{\Pi} = [\mathbf{e}_{n_1}\mathbf{e}_{n_2}\cdots\mathbf{e}_{n_N}]$ in which

Figure 3 Processing pseudo code for the implementation of the proposed T1 detection algorithm for $I_{\max}\geq 1$.

FUNCTION: **Proposed T1 Detection Algorithm**
INPUT: (**H**, **r**, M, N, $|C|$, K, ℓ, I_{max}) OUTPUT: (**s**)

1.	for i = 1 to N, $p_i = \|\mathbf{h}_{:,i}\|^2$ end	Step 1
2.	$\widetilde{\mathbf{H}} = \mathbf{H}\mathbf{\Pi}$, where $\mathbf{\Pi}$ is a $N\times N$ permutation matrix that sorted by p_i from $\mathbf{I}_N$	Step 1
3.	$[\hat{\mathbf{H}}, \hat{\mathbf{r}}]$ = ZF-GIS($\mathbf{H}', \mathbf{H}'', \mathbf{r}, K$), where $\mathbf{H}'$ is first K columns of $\widetilde{\mathbf{H}}$ and $\mathbf{H}''$ is last (N-K) columns of $\widetilde{\mathbf{H}}$	Step 2
4.	$[\mathbf{s}'_{I_1}, \mathbf{s}'_{I_2}, \cdots, \mathbf{s}'_{I_\ell}]$ = B-Chase($\hat{\mathbf{H}}, \hat{\mathbf{r}}, \|C\|, \ell$)	Step 2
5.	$[\mathbf{Q}'', \mathbf{R}'', \mathbf{d}'', \mathbf{\Pi}'']$ = SQRD($\mathbf{H}'', \mathbf{p}''$), where $\mathbf{p}''$ is the 2-norm of each column of $\mathbf{H}''$	Step 3
6.	$\mathbf{v}'' = \mathbf{Q}''^{*}\mathbf{H}'$	Step 3
7.	$\mathbf{u}'' = \mathbf{Q}''^{*}\mathbf{r}$	Step 3
8.	$[\mathbf{Q}', \mathbf{R}', \mathbf{d}', \mathbf{\Pi}']$ = SQRD($\mathbf{H}', \mathbf{p}'$), where $\mathbf{p}'$ is the 2-norm of each column of $\mathbf{H}'$	Step 4
9.	$\mathbf{v}' = \mathbf{Q}'^{*}\mathbf{H}''$	Step 4
10.	$\mathbf{u}' = \mathbf{Q}'^{*}\mathbf{r}$	Step 4
11.	$E_{\min} = \infty$	
12.	for i = 1 to ℓ,	
13.	$I = 0$	
14.	while ($I < I_{\max}$) & ($\mathbf{s}'_{I_i} \neq \mathbf{s}_{I_i}$),	Step 5
15.	if $I \neq 0$, $\mathbf{s}'_{I_i} = \mathbf{s}_{I_i}$ end	Step 5
16.	$\mathbf{y}'' = \mathbf{u}'' - \mathbf{v}''\mathbf{s}'_{I_i}$	Step 3
17.	$\mathbf{s}_{II_i}$ = DF($\mathbf{R}'', \mathbf{d}'', \mathbf{y}'', \mathbf{\Pi}'', N$-$K$)	Step 3
18.	$\mathbf{y}' = \mathbf{u}' - \mathbf{v}'\mathbf{s}_{II_i}$	Step 4
19.	$\mathbf{s}_{I_i}$ = DF($\mathbf{R}', \mathbf{d}', \mathbf{y}', \mathbf{\Pi}', K$)	Step 4
20.	$I = I + 1$	
21.	end	
22.	$\mathbf{e}_i = \mathbf{r} - \mathbf{H}'\mathbf{s}_{I_i} - \mathbf{H}''\mathbf{s}_{II_i}$	Step 6
23.	$\varepsilon_i = 0$	
24.	for j = 1 to M,	
25.	if $\varepsilon_i < E_{\min}$, $\varepsilon_i = \varepsilon_i + \|e_{i,j}\|^2$ end	Step 6
26.	end	
27.	if $\varepsilon_i < E_{\min}$, $E_{\min} = \varepsilon_i$	Step 6
28.	$\widetilde{\mathbf{s}} = [\mathbf{s}_{I_i}\ \mathbf{s}_{II_i}]^T$	Step 6
29.	end	
30.	end	
31.	$\mathbf{s} = \mathbf{\Pi}\widetilde{\mathbf{s}}$	Step 6

$\mathbf{e}_i$ denotes a column vector with zero value for (N-1) elements and the value of one for one element at the i-th order. According to the values of K, $\widetilde{\mathbf{s}}$ can be separated into two group symbols $\mathbf{s}_I = [s_{n_1} s_{n_2} \cdots s_{n_K}]^T$ and $\mathbf{s}_{II} = [s_{n_{K+1}} s_{n_{K+2}} \cdots s_{n_N}]^T$, and $\widetilde{\mathbf{H}}$ can be divided into two sub-channels $\mathbf{H}'$ and $\mathbf{H}''$, where $\mathbf{H}' = [\mathbf{h}_{n_1} \mathbf{h}_{n_2} \cdots \mathbf{h}_{n_K}]$ and $\mathbf{H}'' = [\mathbf{h}_{n_{K+1}} \mathbf{h}_{n_{K+2}} \cdots \mathbf{h}_{n_N}]$.

Step 2: In order to obtain better candidates of Group-I with higher probability, the GIS technique is applied to the system function in (4) and then a lower dimensional sub-system is obtained. We can determine the candidates of Group-I from the sub-system by smaller ED. On the other hand, we modify the original ZF-GIS computation [15] to lower the complexity in Fig. 4. Without loss of

FUNCTION: **ZF-GIS**
INPUT: $(\mathbf{H}', \mathbf{H}'', \mathbf{r}, K)$ OUTPUT: $(\hat{\mathbf{H}}, \hat{\mathbf{r}})$

1. $\mathbf{H}_b$ = $(N\text{-}K)\times(N\text{-}K)$ matrix at the bottom of $\mathbf{H}''$
2. $\mathbf{W} = (\mathbf{H}_b^T)^{-1}$
3. for i = 1 to $(M\text{-}N\text{+}K)$, $\mathbf{x}_i = -\mathbf{W}\mathbf{h}_{i,:}''^T$ end
4. $\mathbf{X} = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_{M-N+K}^T]^T$
5. $\mathbf{Z} = [\mathbf{I}_{M\text{-}N\text{+}K} \ \vdots \ \mathbf{X}]$
6. $\mathbf{L}$ = GSO($\mathbf{Z}$)
7. $\hat{\mathbf{H}} = \mathbf{L}\mathbf{H}'$
8. $\hat{\mathbf{r}} = \mathbf{L}\mathbf{r}$

Figure 4 Processing pseudo code for the proposed modified GIS implementation.

the generality, the ordered channel matrix $\widetilde{\mathbf{H}}$ can be written as

$$\widetilde{\mathbf{H}} = [\mathbf{H}'\mathbf{H}''] = [\mathbf{h}_{n_1} \ \mathbf{h}_{n_2} \ \cdots \ \mathbf{h}_{n_K} \ \mathbf{h}_{n_{K+1}} \ \cdots \ \mathbf{h}_{n_N}] = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,K} & h_{1,K+1} & \cdots & h_{1,N} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,K} & h_{2,K+1} & \cdots & h_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{M,1} & h_{M,2} & \cdots & h_{M,K} & h_{M,K+1} & \cdots & h_{M,N} \end{bmatrix}, \quad (5)$$

In the modified ZF-GIS, we employ the matrix $\mathbf{H}_b$ to obtain a left null matrix $\mathbf{Z}$ of $\mathbf{H}''$, where $\mathbf{H}_b$ is an $(N\text{–}K)\times(N\text{–}K)$ square matrix on the bottom of $\mathbf{H}''$and $\mathbf{Z}$ is an $(M - N\text{+}K)\times M$ matrix. $\mathbf{H}_b$ and $\mathbf{Z}$ can be respectively expressed as

$$\mathbf{H}_b = \begin{bmatrix} h_{M-N+K+1,K+1} & h_{M-N+K+1,K+2} & \cdots & h_{M-N+K+1,N} \\ h_{M-N+K+2,K+1} & h_{M-N+K+2,K+2} & \cdots & h_{M-N+K+2,N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M,K+1} & h_{M,K+2} & \cdots & h_{M,N} \end{bmatrix}, \quad (6)$$

and

$$\mathbf{Z} = [\mathbf{I}_{M-N+K}\mathbf{X}] = \begin{bmatrix} 1 & 0 & \cdots & 0 & x_{1,1} & x_{1,2} & \cdots & x_{1,N-K} \\ 0 & 1 & \cdots & 0 & x_{2,1} & x_{2,2} & \cdots & x_{2,N-K} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & x_{M-N+K,1} & x_{M-N+K,2} & \cdots & x_{M-N+K,N-K} \end{bmatrix}, \quad (7)$$

We define $\mathbf{x}_i$=$[x_{i,1} \ x_{i,2} \ \dots x_{i,(N\text{-}K)}]^T$, and $\mathbf{x}_i$ can be calculated via the following matrix computation:

$$\mathbf{x}_i = -\left(\mathbf{H}_b^T\right)^{-1}\mathbf{h}''^T_{i,:} \text{ for } i = 1, 2, \dots, (M - N + K), \quad (8)$$

where $\mathbf{h}''_{i,:}$ denotes the i-th row of $\mathbf{H}''$. In this way, we can retrieve the left null matrix $\mathbf{Z}$ and then apply the Gram-Schmidt orthogonalization [25] to $\mathbf{Z}$ to obtain a row-orthogonal matrix $\mathbf{L}$. The original ZF-GIS [14, 15] needs to compute $\overline{\mathbf{Z}} = \mathbf{I}_M - \mathbf{H}''\left(\mathbf{H}''^*\mathbf{H}''\right)^{-1}\mathbf{H}''^*$ first, where $\mathbf{x}^*$ denotes the conjugate transpose of $\mathbf{x}$, Next, $(M\text{-}N\text{+}K)$ row vectors are selected from $\overline{\mathbf{Z}}$ to obtain $\mathbf{Z}$. That means the computation for $(N\text{-}K)$ row vectors are unnecessary. After ZF-GIS, $\mathbf{L}$ is multiplied on both sides of (4) and we can derive the following sub-system as

$$\hat{\mathbf{r}} = \hat{\mathbf{H}}\mathbf{s}_I' + \hat{\mathbf{n}}, \quad (9)$$

where $\hat{\mathbf{n}} = \mathbf{L}\tilde{\mathbf{n}}$ and $\hat{\mathbf{H}} = \mathbf{L}\mathbf{H}'$ with dimension of $(M\text{–}N\text{+}K)\times K$. After the ZF-GIS operation, we use the B-Chase detection algorithm [6] as sa_1 to detect the sub-system in (9) by choosing better ℓ candidates with smaller ED, where ℓ ranges from 1 to $|C|$. Then, we can derive an ordered list of partial candidates $\left\{\mathbf{s}_{I_1}', \mathbf{s}_{I_2}', \cdots, \mathbf{s}_{I_\ell}'\right\}$ with ℓ least-ED candidates in this sub-system.

Steps 3, 4, and 5: For convenience of illustration, the operations in Steps 3, 4 and 5 are jointly described together. We just describe the operation of the i-th iterative decision feedback (IDF). In Steps 3 and 4 of the proposed work, we apply the SQRDF algorithm as sa_2 and sa_3 to detect two sub-systems in (10) and (11):

$$\mathbf{r}_i' = \mathbf{r} - \mathbf{H}'\mathbf{s}_{I_i}' = \mathbf{H}''\mathbf{s}_{II_i} + \mathbf{n}', \quad (10)$$

$$\mathbf{r}_i'' = \mathbf{r} - \mathbf{H}''\mathbf{s}_{II_i} = \mathbf{H}'\mathbf{s}_{I_i} + \mathbf{n}'', \quad (11)$$

The SQRDF algorithm can be divided into two parts: sorted QR decomposition (SQRD) and decision feedback (DF) whose pseudo code is listed in Fig. 5. Both parts adopt the algorithm in [19]. After the SQRD operation on $\mathbf{H}''$, we can derive $\mathbf{H}''\mathbf{\Pi}'' = \mathbf{Q}''\mathbf{R}''$, where $\mathbf{Q}''$, $\mathbf{R}''$, $\mathbf{\Pi}''$ denote the unitary matrix, upper triangular matrix with positive and real diagonal elements, and permutation matrix, respectively. Next, we can obtain the vector $\mathbf{d}''$ which contains the

FUNCTION: **DF**
INPUT: $(\mathbf{R}, \mathbf{y}, \mathbf{d}, \mathbf{\Pi}, J)$ OUTPUT: $(\mathbf{s})$

1. for i = 1 to J
2. $t = \sum_{j=1}^{i-1} r_{i,j}\bar{s}_j$
3. $\bar{s}_i = \mathbf{quan}((y_i - t)d_i)$
4. end
5. $\mathbf{s} = \mathbf{\Pi}\bar{\mathbf{s}}$

Figure 5 Processing pseudo code for the DF implementation.

reciprocal of the diagonal elements of $\mathbf{R}''$. After multiplying $\mathbf{Q}''^{*}$ on both sides of (10), the system function can be changed to

$$\mathbf{y}_i'' = \mathbf{Q}''^{*}\mathbf{r}_i'' = \mathbf{Q}''^{*}\mathbf{r} - \mathbf{Q}''^{*}\mathbf{H}'\mathbf{s}_{I_i}' = \mathbf{R}''\overline{\mathbf{s}}_{II_i} + v'', \tag{12}$$

where $\overline{\mathbf{s}}_{II_i} = \mathbf{\Pi}''^{*}\mathbf{s}_{II_i} = \left[\overline{s}_{i,n_{K+1}}\overline{s}_{i,n_{K+2}}\cdots\overline{s}_{i,n_N}\right]^T$. $\overline{\mathbf{s}}_{II_i}$ can be obtained by the DF operation as shown in Fig. 5, where the elements of $\overline{\mathbf{s}}_{II_i}$ can be expressed as

$$\overline{s}_{i,n_b} = \mathbf{quan}\left(\left(\mathbf{y}_{i,b-K}'' - \sum_{j=b-K+1}^{N-K}\mathbf{R}_{b-K,j}''\overline{s}_{i,n_{j+K}}\right)d_{b-K,b-K}''\right)$$
$$\text{for } b = N, N-1, \ldots, K+1 \tag{13}$$

where **quan**(x) denotes the quantization function that quantizes the value x to the nearest constellation point. The symbols $\mathbf{s}_{II_i}$ can be obtained by reordering $\overline{\mathbf{s}}_{II_i}$. Similarly, in Step 4, we can obtain following equations:

$$\mathbf{y}_i' = \mathbf{Q}'^{*}\mathbf{r}_i'' = \mathbf{Q}'^{*}\mathbf{r} - \mathbf{Q}'^{*}\mathbf{H}''\mathbf{s}_{II_i} = \mathbf{R}'\overline{\mathbf{s}}_{I_i} + v', \tag{14}$$

$$\overline{s}_{i,n_c} = \mathbf{quan}\left(\left(\mathbf{y}_{i,K-c+1}' - \sum_{j=K-c+2}^{K}\mathbf{R}_{K-c+1,j}'\overline{s}_{i,n_{K-j+1}}\right)d_{K-c+1,K-c+1}'\right)$$
$$\text{for } c = 1, 2, \ldots, K, \tag{15}$$

where $\mathbf{H}'\mathbf{\Pi}' = \mathbf{Q}'\mathbf{R}'$ and $\overline{\mathbf{s}}_{I_i} = \mathbf{\Pi}'^{*}\mathbf{s}_{I_i} = \left[\overline{s}_{i,n_K}\overline{s}_{i,n_{K-1}}\cdots\overline{s}_{i,n_1}\right]^T$. The maximum iteration number $I_{\max}$ affects the computational complexity and performance. The initial iteration number I is set to zero. When executing Step 4 once, I is increased by one. If $\mathbf{s}_{I_i}'$ equals $\mathbf{s}_{I_i}$ or I equals $I_{\max}$, the candidate $\widetilde{\mathbf{s}}_{n_i} = [\mathbf{s}_{I_i}\ \mathbf{s}_{II_i}]^T$ is obtained. Otherwise, let $\mathbf{s}_{I_i}' = \mathbf{s}_{I_i}$ and repeat Steps 3 and 4. Note that if $I_{\max}$=0, there is no need to deal with the sub-system in (11), and the operations in (14) and (15) can be skipped.

Step 6: At the last step, we choose the final hard decision $\widetilde{\mathbf{s}}$ according to the MED criterion among the candidates $\{\widetilde{\mathbf{s}}_{n_1}, \widetilde{\mathbf{s}}_{n_2}, \cdots, \widetilde{\mathbf{s}}_{n_\ell}\}$. The ED of the i-th candidate is obtained by $\varepsilon_i = ||\mathbf{r} - \widetilde{\mathbf{H}}\widetilde{\mathbf{s}}_i||^2$. According to the permutation matrix $\mathbf{\Pi}$ in Step 1, we rank the detected symbols $\widetilde{\mathbf{s}}$ to obtain the final symbols $\mathbf{s}$.

There are three schemes to lower the computational complexity for T1 algorithm. First, the GIS technique is employed one time to reduce the tree search (TS) complexity by choosing fewer candidates under satisfactory BER performance. On the other hand, we can reduce PP complexity since the lower dimensional sub-matrix is processed. Second, we reduce PP complexity by reusing tentative computations.

- Observing (12) and (14), we can reuse tentative calculations for parallel and iterative computation such that we just compute the SQRD function on $\mathbf{H}'$ and $\mathbf{H}''$, $\mathbf{Q}''^{*}\mathbf{r}$, $\mathbf{Q}''^{*}\mathbf{H}'$, $\mathbf{Q}'^{*}\mathbf{r}$, and $\mathbf{Q}'^{*}\mathbf{H}''$ once.
- Observing (3), $\mathbf{p}' = [p_{n_1}p_{n_2}\cdots p_{n_K}]^T$ and $\mathbf{p}'' = \left[p_{n_{K+1}}\ p_{n_{K+2}}\cdots p_{n_N}\right]^T$ can be reused in the computation of the SQRD function.

Third, we reduce TS complexity by avoiding unnecessary computations.

- When the input symbols are the same as that of the previous iteration, the calculations in the following iterations can be skipped. That means we do not need to reach the maximum iteration number $I_{\max}$ in each IDF.
- The pruning and threshold-tightening strategy given in [6] is used to generate a threshold $E_{\min}$ which records the MED value of other previous candidates. When the ED is greater than $E_{\min}$ during the computation, the process can be terminated.

Using the above three schemes, the computational complexity can be alleviated and the block diagram of the proposed T1 algorithm is shown in Fig. 6.

4.2 Type 2 (T2) Detection Algorithm

Recently, a grouped detection algorithm with scalable property [23] has been proposed to lower the computational complexity by using 2x2 V-BLAST with ML (V-ML) scheme and herein, we called this scheme as scalable GD (S-GD) algorithm. Although the S-GD algorithm can result in lower complexity, while the number of antennas increases, the larger degradation of the BER performance is incurred because the system applies the GIS technique multiple times. On the other hand, although the T1 algorithm possesses a better trade-off in items of complexity and performance, the reduction of complexity is still confined. Thus, the T2 algorithm is proposed to achieve the lower computational complexity via the proposed framework. In the proposed T2 algorithm, the parameters of $\{K, \ell, I_{\max}\}$ are set to {2, 1, 0} and 2x2 V-ML scheme [23] is adopted as the sa_1 in Step 2. Other two sub-algorithms and the processing steps of the T2 algorithm are the same as that of the T1 algorithm. In this manner, the complexity of PP and TS can be further alleviated. Owing to the above parameter and sub-algorithm setting, the T2 algorithm can achieve much lower complexity than that of the T1

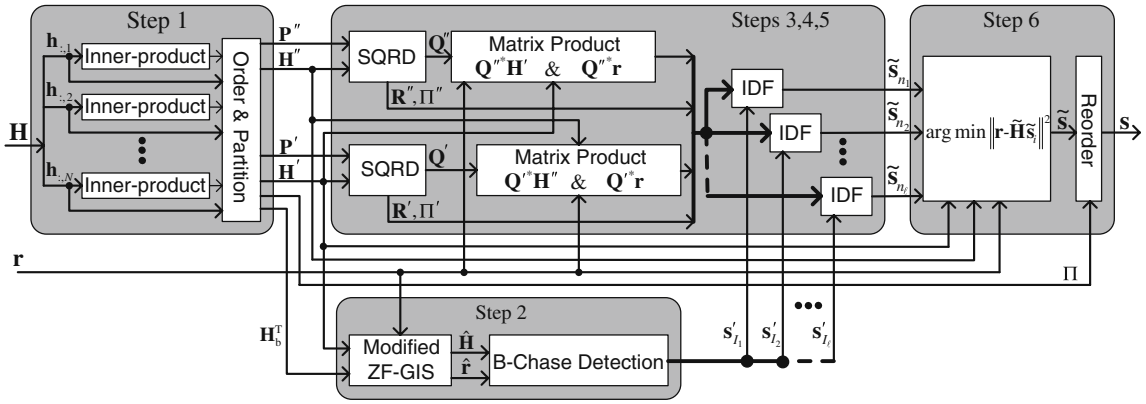

Figure 6 Block diagram of the proposed T1 algorithm.

algorithm at penalty of the loss of the BER performance. Compared with the S-GD and BODF algorithms, the T2 algorithm can result in better BER performance with lower complexity. In the following section, the complexity analysis of the T1 and T2 algorithms will be discussed in detail.

5 Discussion and Simulation Results

This section demonstrates the complexity and performance of the T1 and T2 detection algorithms and shows the comparison results with the existing detection schemes including the BODF, S-GD, B-Chase and GPIC(K,0) detection algorithms. We use the T1(K, ℓ, $I_{\max}$) to denote the T1 algorithm with K symbols distributed to Group-I, list length ℓ and maximum iteration $I_{\max}$. Moreover, the B-Chase(ℓ) denotes the ZF B-Chase algorithm with list length ℓ, and the GPIC(K,E) denotes the GPIC algorithm with K symbols in Group-I and E error symbols in Group-II.

5.1 Parallel Characteristic Comparison

In Table 2, the parallel characteristic comparison in qualitative way among the existing parallel MIMO detection algorithms is presented. Since the T2, BODF and S-GD algorithms do not belong to the parallel detection algorithm, we do not include the above three algorithms in Table 2. The T1 algorithm has a wide range of the number of parallel symbols like GPIC algorithm in [7] and the list algorithm in [21], where the number of parallel symbols denotes the number of symbols in Group-I. In Table 2, the maximum signal power is obtained by computing 2-norms of all columns of channel matrix and the minimum SNR or selection algorithm 1 and 2 in [6] are obtained by calculating the QR decomposition or pseudo inverse of channel matrix. It is known that the former leads to less computational complexity. Since the T1 algorithm chooses the parallel symbols with maximum signal power in (3) and other parallel algorithms either use the symbols with minimum SNR or selection algorithm 1 and 2 in [6], the PP complexity of the T1 algorithm can be significantly saved. The T1 algorithm parallelizes the computation by choosing candidates with smaller ED. Other parallel detection algorithms cannot apply this scheme because the ED cannot be retrieved without total symbols. Due to the new feature of choosing candidates with smaller ED, the T1 algorithm is capable of choosing better or fewer candidates than other parallel detection algorithms [6, 7, 20, 21] under the limited complexity. On the other hand, because of fewer

Table 2 Parallel characteristic comparison of the MIMO detection algorithms.

Algorithm	Number of parallel symbols: K	How to choose parallel symbols	How to parallelize parallel symbols	How to detect residual symbols
Parallel [20]	$K=1$	Selection algorithm 1 in [6]	Fully expanded	Any detection algorithm
List [21]	$1\le K<N$	Minimum SNR	Choose adjacent points after linear detection	BODF
B-Chase [6]	$K=1$	Selection algorithm 1 or 2 in [6]	Choose adjacent points after linear detection	BODF
GPIC [7]	$1\le K<N$	Minimum SNR	Fully expanded	Redetection scheme in [7]
Proposed T1	$1<K<N$	Maximum signal power	Choose candidates with smaller ED after B-Chase detection	Iterative & SQRDF

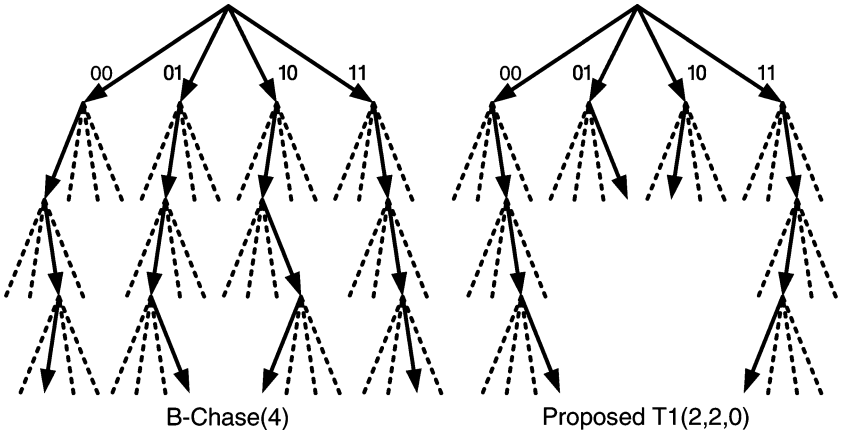

Figure 7 Searching tree diagrams of the B-Chase(4) and proposed T1(2,2,0) in four transmitters with QPSK inputs.

candidates, the T1 algorithm reduces the computational complexity of detecting residual symbols and ED. For example, the searching tree diagrams of the B-Chase(4) and T1(2,2,0) in four transmitters with QPSK inputs are depicted in Fig. 7, where the solid line represents the path that has been searched. Note that the detection order of B-Chase(4) and T1(2,2,0) can be different. From Fig. 7, the T1(2,2,0) has fewer solid-line paths than B-Chase(4) because the T1(2,2,0) only chooses two candidates from the first two symbols.

5.2 Numerical Complexity Comparison

In Table 3, the number of complex multiplications, complex divisions and square roots required by the T1 algorithm are listed. The T1 algorithm includes the following functions: the order and partition symbols (OPS), GIS, B-Chase used in sub-system, precomputation1 (PC1), precomputation2 (PC2) and the combination of DF and MED (DF&MED) of the design steps. The complexity expression of PP and/or TS of each function is tabulated in Table 3. PC1(1) and PC1(2) correspond to the operations of lines 5–6 and 7 of Fig. 3, respectively. Similarly, PC2(1) and PC2(2) represent the operations of lines 8–9 and 10 of Fig. 3, respectively. Although division and square root computations are more complicated than multiplication, the number of divisions and square roots is much less than the number of multiplications in T1 and others algorithms. Moreover, since the multiplication dominates the computational complexity in the system, the complexity is measured by the sum of complex multiplications, divisions and square roots rather than additions in the worst case. Note that the

Table 3 Computational complexity of the proposed T1 algorithm.

	Function belong to	Complexity Belong to	Multiplications	Divisions	Square roots
OPS	Step 1	PP	MN	0	0
GIS	Step 2	PP	$1/6M^3+1/2M^2N-1/2MN^2+MNK-1/2MK^2+3/2N^3-5N^2K+11/2NK^2-2K^3+M^2+3/2MK-MN+5/2N^2-11/2NK+3K^2-1/6M-N+K$	$M+N-K$	M
		TS	$1/2M^2-1/2N^2+NK-1/2K^2+1/2M-1/2N+1/2K$	0	0
B-Chase	Step 2	PP	$2MK^2-2NK^2+11/3K^3+MK-NK+9/2K^2-7/6K$	$3K$	$2K$
		TS[a]	$MK-NK+K^2+2K\|C\|$	0	0
PC1 (1)	Step 3	PP	$MN^2-MNK+1/2N^2-NK+1/2K^2-1/2N+1/2K$	$N-K$	$N-K$
(2)		TS	$MN-MK$	0	0
PC2 (1)	Step 4	PP	$MNK+1/2K^2-1/2K$	K	K
(2)		TS	MK	0	0
DF&MED[b]	Steps 3, 4, 6	TS	$(NI_{\max}+M)\ell$	0	0
Proposed T1total	All steps	PP	$1/6M^3+1/2M^2N+1/2MN^2+MNK+3/2MK^2+3/2N^3-5N^2K+7/2NK^2+5/3K^3+M^2+5/2MK+3N^2-15/2NK+17/2K^2-1/6M-3/2N-1/6K$	$M+2N+2K$	$M+N+2K$
		TS	$1/2M^2+MN-1/2N^2+MK+1/2K^2+1/2M-1/2N+1/2K+2K\|C\|+(NI_{\max}+M)\ell$	0	0

[a] When ℓ equals $|C|$, the TS computational complexity of sub-algorithm B-Chase is changed to $MK-NK+K^2$

[b] When $I_{\max}$ equals zero, the complexity of PC2 is equal to zero and the multiplication complexity of DF&MED is changed to $(M+N-K)\ell$

Table 4 Complexity comparison among the proposed and conventional algorithms.

Algorithm	Type	Multiplications/Divisions/Square roots
B-Chase	PP[a]	$2MN^2+5/3N^3+MN+7/2N^2+23/6N$
	TS	$MN+2N\ell$
GPIC(1,0)	PP	$4MN^2-4MN+N^2+3/2M-2N+1$
	TS	$MN\|C\|$
Proposed T1[b]	PP	$1/6M^3+1/2M^2N+11/8MN^2+1/12N^3+M^2+5/4MN+11/8N^2+11/6M+41/12N$
	TS	$1/2M^2+3/2MN-3/8N^2+1/2M-1/4N+N\|C\|+(NI_{max}+M)\ell$
Proposed T2	PP	$1/6M^3+1/2M^2N+1/2MN^2+3/2N^3+M^2-7N^2+17/6M+21/2N$
	TS	$1/2M^2+MN-1/2N^2-1/2M+5/2N-3+4\|C\|$

[a] When $1<\ell<|C|$, the additional computation complexity is needed.

[b] In this case, N is an even integer, $K=N/2$, $\ell<|C|$ and $I_{max}\geq 1$.

multiplication of a number and a constellation point can be implemented by scaled integers [26] such that the multiplication can be realized by shift and addition. For simplicity, we assume that the number of transmitters is an even integer and $K=N/2$ in the T1 algorithm. The comparisons of PP and TS computational complexity of the T1, T2, B-Chase, and GPIC(1,0) algorithms are tabulated in Table 4. When $M=N$, the complexity order of PP of the T1, T2, B-Chase and GPIC algorithms are $O(17/8N^3)$, $O(8/3N^3)$, $O(11/3N^3)$ and $O(4N^3)$, respectively. That means the T1 and T2 algorithms have lower PP complexity than other detection algorithms.

Herein, we do not formulate the complexity of the GD and ID algorithms since both algorithms require more computational complexity than the B-Chase detection, where the complexity of the GD algorithm is exponential growth by the number of Group-I symbols and the complexity of the ID algorithm almost doubles that of the BODF algorithm mentioned in [18]. The SD detector shows larger computational complexity as addressed in [6], for example, at BER=10^{-3}, the SD and B-Chase algorithms respectively own the complexity of 57 RM/b and 18 RM/b, where RM/b represents the required number of real multiplications per detected bit. Thus, we consider the B-Chase detection algorithm for complexity comparison instead of the GD, ID, and SD algorithms.

Furthermore, concerning the influence of changes of the channel matrix every T symbol period, the complex multipliers per bit (CMPB) is defined as follows.

$$\mathrm{CMPB}=\frac{\mathrm{TS}+\mathrm{PP}/T}{N\cdot\log_2|C|}, \tag{16}$$

The total complexity of T1, T2, S-GD, B-Chase, and GPIC(1,0) algorithms are calculated by (16) in the following two subsections.

5.3 Simulation Results

The simulation environment is assumed Rayleigh fading channel and no correlation between sub-channels. The SNR is defined as the signal power over the noise power in the

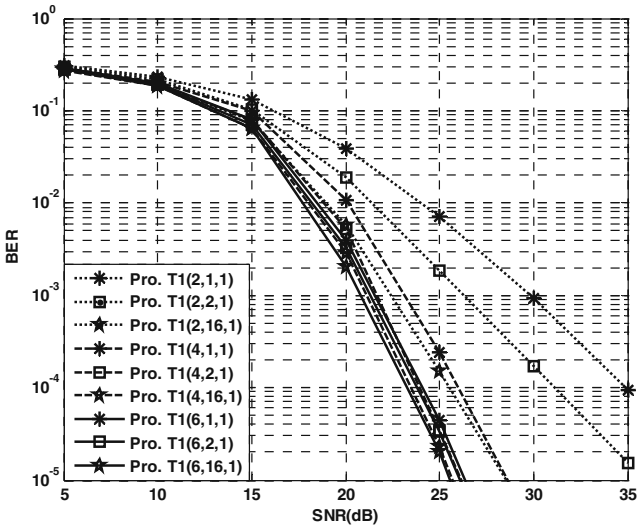

Figure 8 BER performance of the proposed T1 algorithm with different K and ℓ in (8,8) MIMO system with 16-QAM inputs.

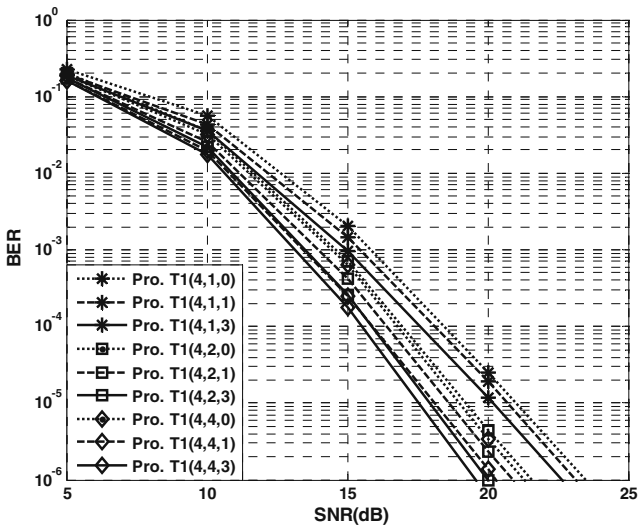

Figure 9 BER performance of the proposed T1 algorithm with different I_{max} and ℓ in (8,8) MIMO system with QPSK inputs.

Table 5 Average complexity result of the proposed T1 algorithm by skipping unnecessary iteration in (8,8) MIMO system with 16-QAM inputs.

Algorithm	I_{max}	SNR(dB)					
		10	15	20	25	30	35
Pro. T1(4,1,3)	3	1.71(57.0%)	1.37(45.7%)	1.04(34.7%)	1.00(33.3%)	1.00(33.3%)	1.00(33.3%)
Pro. T1(4,2,3)	3	1.77(59.0%)	1.49(49.7%)	1.32(44.0%)	1.31(43.7%)	1.32(44.0%)	1.32(44.0%)
Pro. T1(4,4,3)	3	1.85(61.7%)	1.65(55.0%)	1.54(51.4%)	1.51(50.4%)	1.52(50.7%)	1.52(50.7%)
Pro. T1(4,16,3)	3	2.23(74.4%)	2.14(71.4%)	2.10(70.0%)	2.09(69.7%)	2.09(69.7%)	2.09(69.7%)

receiver, where the noise is white Gaussian random variable with zero mean. The performance measurement targets at the SNR with BER=10^{-3}. Figure 8 shows the performance of the T1 algorithm with different K and ℓ in (8,8) system with 16-QAM inputs. We can find that the performance with larger K is better than that with smaller K under the same ℓ. In this case, the complexity of the T1 algorithm with K=6 approximately doubles with K=2 under the same ℓ with T=1, where T denotes the symbol period for the changes of the channel matrix, and the range of performance of the T1 algorithm with K=6 is narrow. In order to trade off the complexity and performance, choosing K in the range from 2 to N/2 is preferred. Figure 9 shows the performance of the T1 algorithm with different I_{max} and ℓ in (8,8) system with QPSK inputs. The performance of the T1 algorithm can be improved by increasing I_{max} under the same ℓ. For example, T1(4,4,1) outperforms T1(4,4,0) by 1 dB and T1(4,4,3) outperforms T1(4,4,1) by 0.3 dB. Therefore, if the complexity constraint is not critical, the I_{max} could be further enlarged. On the other hand, if the low complexity is demanded, the I_{max} could be zero for the satisfactory BER performance. We set K=N/2 and I_{max}=1 in the T1 algorithm to compare with the existing detection algorithms. Considering TS complexity reduced by avoiding unnecessary computation scheme, we show the average complexity results by skipping unnecessary iteration and pruning and threshold-tightening strategies [6] in Tables 5 and 6, respectively. On average, for SNR=20 dB, the iteration reduction by T1(4,4,3) can be up to 48.6%.

Figures 10, 11, 12 and 13 show the performance in (4,4) and (8,8) systems, where Figs. 10 and 12 use the constellation of QPSK, and Figs. 11 and 13 use the constellation of 16-QAM. From the simulation results, we can find out the BER performance of the T1 algorithm can be significantly enhanced by slightly increasing the list length ℓ. For example, T1(2,2,1) outperforms T1(2,1,1) by 3.3 dB and 3 dB with respect to QPSK and 16-QAM inputs in (4,4) system, and just increases complexity by 10.3% and 6.4% when T=8. The better performance can be obtained with longer list length. For example, T1(2,16,1) outperforms T1(2,1,1) by 5 dB with 16-QAM inputs. In summary, the computational complexity and BER performance of the T1 algorithm depends on these parameters given above. The smaller K, ℓ, I_{max}, and simplified sub-algorithm achieve lower complexity. Otherwise, the higher K, ℓ, I_{max}, and better sub-algorithm attain better performance. Observing the BER performance simulations, it can be summarized that ℓ=2 in QPSK modulation and ℓ=4 in 16-QAM modulation are reasonable setting with K=N/2 for the T1 algorithm. On the other hand, in (8,8) system, compared with the S-GD algorithm, the T2 algorithm not only results in the complexity reduction of 26.6% and 23.1% when T=8 but also outperforms 2.9 dB and 3.2 dB with respect to QPSK and 16-QAM inputs. In (8,8) system with 16-QAM inputs,

Table 6 Average complexity result of the proposed T1 algorithm by pruning and threshold-tightening strategy [6] in (8,8) MIMO system with 16-QAM inputs.

Algorithm	E_{max}	SNR(dB)					
		10	15	20	25	30	35
Pro. T1(4,2,1)	16	14.95(93.4%)	14.04(87.7%)	11.29(70.6%)	10.01(62.6%)	9.66(60.4%)	9.57(59.8%)
Pro. T1(4,4,1)	32	27.26(85.2%)	24.17(75.5%)	16.92(52.9%)	13.85(43.3%)	12.92(40.4%)	12.65(39.5%)
Pro. T1(4,16,1)	128	83.65(65.4%)	66.98(52.3%)	41.20(32.2%)	30.56(23.9%)	27.16(21.2%)	26.14(20.4%)
Pro. T1(4,2,3)	16	15.26(95.4%)	14.48(90.5%)	12.31(76.9%)	11.37(71.1%)	11.19(69.9%)	11.09(69.3%)
Pro. T1(4,4,3)	32	28.64(89.5%)	26.11(81.6%)	20.36(63.6%)	18.18(56.8%)	17.76(55.5%)	17.53(54.8%)
Pro. T1(4,16,3)	128	99.88(78.0%)	85.16(66.5%)	59.02(46.1%)	49.31(38.5%)	45.94(35.9%)	45.14(35.3%)

* E_{max}=M×ℓ

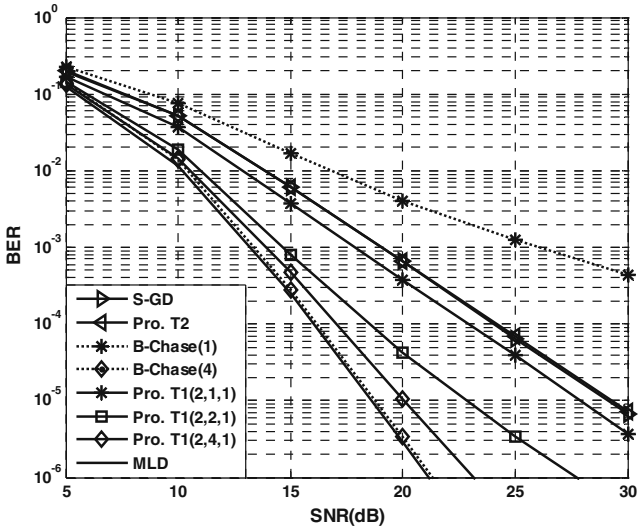

Figure 10 BER performance of the proposed T1, proposed T2 and conventional algorithms in (4,4) MIMO system with QPSK inputs.

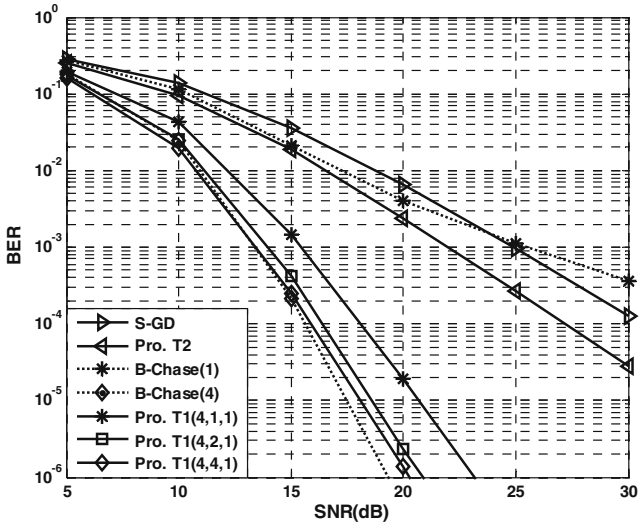

Figure 12 BER performance of the proposed T1, proposed T2 and conventional algorithms in (8,8) MIMO system with QPSK inputs.

the T2 algorithm outperforms the BODF algorithm by 3.1 dB and reduces complexity by 21.9% when T=8.

5.4 Complexity and Performance Tradeoff

The complexity-performance trade off of the T1, T2, S-GD, B-Chase, and GPIC(1,0) algorithms is shown in Figs. 14 and 15, where the performance is measured by the SNR with BER=10^{-3} and the complexity is measured with T=8. In (8,8) system, T1(4,1,1) with QPSK and 16-QAM inputs gains 10 dB and 9.9 dB, respectively, compared with the BODF algorithm (B-Chase(1)) and leads to the complexity reduction by 8.5% and −18.5%, respectively. T1(4,16,1), T1(4,4,1), and T1(4,2,1) reduce complexity by 10.5%, 21.2%, and 26.6% while falling 0.5 dB, 0.6 dB, and 1 dB short of the B-Chase(16) algorithm with 16-QAM inputs, respectively. In other configurations with M=N, the comparison of complexity and performance has behavior similar to that of the above analysis trend.

Figure 16 shows the comparison results of the complexity ratio between some pairs of the T1, T2, S-GD, B-Chase, and GPIC(1,0) algorithms versus T. T1(4,4,1) falls 0.6 dB compared with B-Chase(16) while reducing complexity by at least 7% for T<8192. T1(4,1,1) outperforms B-Chase(12) by 0.1 dB and results in the complexity reduction by at least 11.5% for T<32. T1(4,16,1) not only outperforms GPIC(1,0)

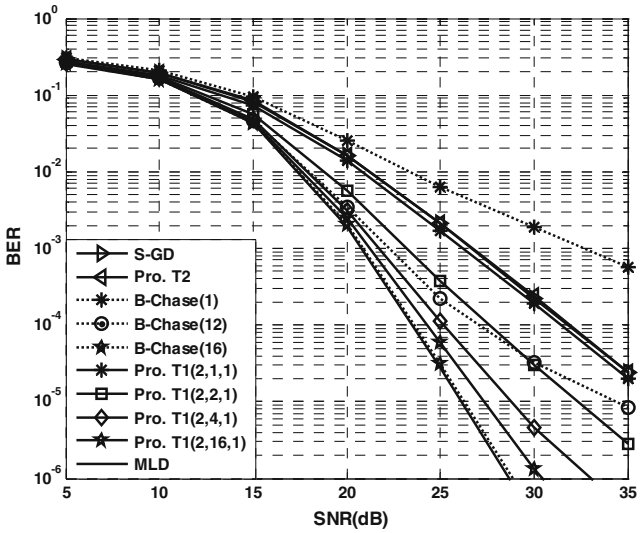

Figure 11 BER performance of the proposed T1, proposed T2 and conventional algorithms in (4,4) MIMO system with 16-QAM inputs.

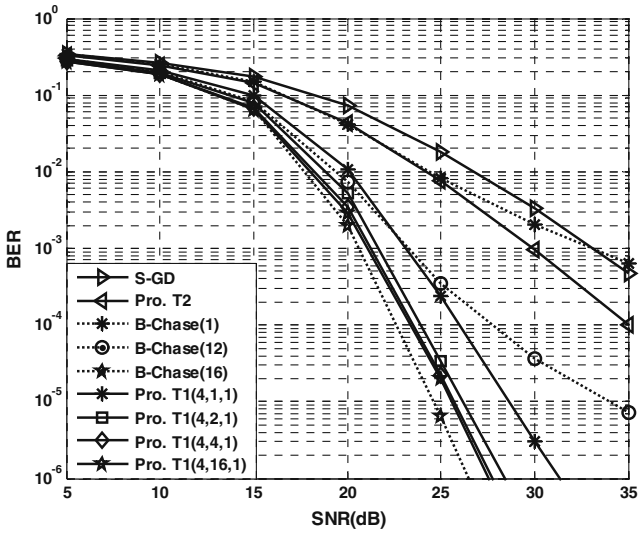

Figure 13 BER performance of the proposed T1, proposed T2 and conventional algorithms in (8,8) MIMO system with 16-QAM inputs.

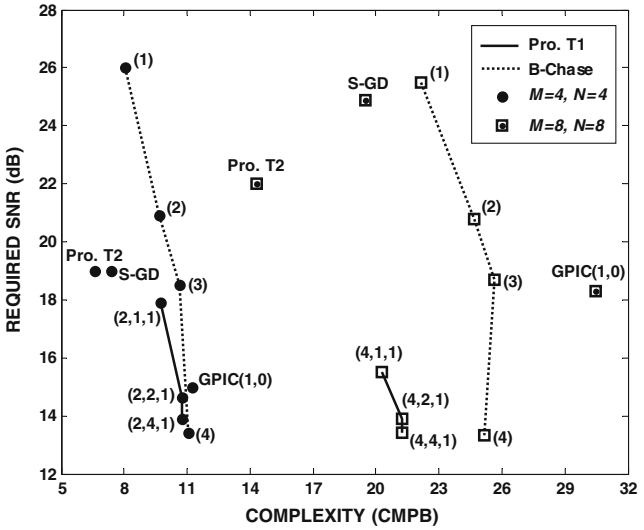

Figure 14 Complexity-performance tradeoff of the proposed T1, proposed T2, S-GD, B-Chase and GPIC(1,0) algorithms with QPSK inputs and T=8.

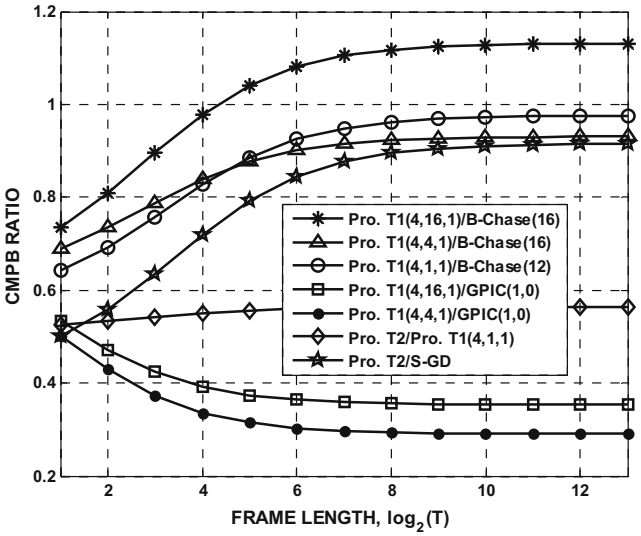

Figure 16 Complexity ratio of the proposed T1, proposed T2, S-GD, B-Chase, and GPIC(1,0) algorithms in (8,8) MIMO system with 16-QAM inputs.

by 4.1 dB but also reduces complexity by at least 50% for T> 4. T1(4,16,1) falls 0.5 dB short of the B-Chase(16) while reducing complexity by 26.4% when T=1. However, the CMPB ratio is larger than one but less than 1.2 when T>32. Considering the comparison between the T2 and T1(4,1,1) algorithms, T2 reduces complexity by at least 43.6% for T< 8192. The T2 outperforms S-GD by 3.2 dB with the complexity reduction by at least 8.5% for T<8192. Therefore, from the complexity and performance analysis, the T1 algorithm attains better complexity-performance tradeoff at the slight penalty of BER performance degradation compared with the B-Chase and GPIC(1,0) detection algorithms. Also, compared with the S-GD, BODF and T1 algorithms, the T2 algorithm can attain lower computational complexity. From above simulation results, the T1 and T2 algorithms are efficient in terms of complexity-performance tradeoff.

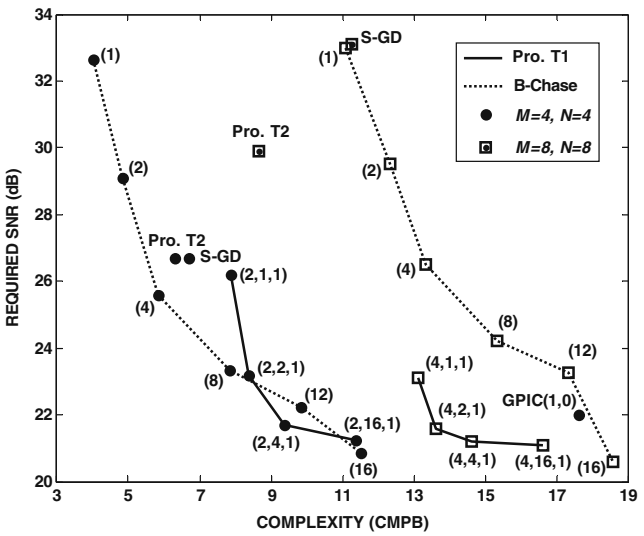

Figure 15 Complexity-performance tradeoff of the proposed T1, proposed T2, S-GD, B-Chase and GPIC(1,0) algorithms with 16-QAM inputs and T=8.

6 Conclusions

In this paper, the proposed generalized framework is capable of generating six conventional and two new efficient MIMO detection algorithms. Due to the GIS technique, the T1 algorithm can choose better or fewer candidates with higher probability than other parallel detection algorithms under the limited complexity. On the other hand, the PP complexity of the T1 and T2 algorithms can be reduced since the lower dimensional sub-matrix is processed after applying the GIS technique. The T1 detection algorithm attains better complexity-performance tradeoff than the B-Chase detection algorithm with slightly sacrificing BER performance. For example, in (8,8) system with 16-QAM inputs, at high performance end, T1(4,4,1) and T1(4,2,1) can reduce multiplication complexity by 21.2% and 26.6% at the penalty of 0.6 dB and 1 dB loss compared with the B-Chase(16) detection, respectively. Furthermore, the proposed T2 algorithm can attain lower complexity compared with the T1, S-GD, BODF, B-Chase algorithms. For example, in (8,8) system with 16-QAM inputs, the T2 algorithm not only reduces complexity by 21.9% but also outperforms 3.1 dB compared with the BODF detection algorithm.

Acknowledgement This work was supported in part by the National Science Council (NSC) Grant NSC-98-2220-E-009-042, NSC-97-2220-E-009-024.

Appendix

Table 7 Glossary of acronym defined in this paper

Acronym	Definition
CMPB	complex multipliers per bit
DF	decision feedback
ED	Euclidean distance
GIS	group interference suppression
MED	minimum Euclidean distance
OPS	order and partition symbols
PC1	precomputation1
PC2	precomputation2
PIC	parallel interference cancellation
PP	preprocessing
SQRD	sorted QR decomposition
TS	tree search

References

1. Foschini, G. J. (1996). Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Technical Journal, 1*(2), 41–59.
2. Telatar, I. E. (1999). Capacity of multi-antenna Gaussian channels. *European Transactions On Telecommunications, 10*(6), 585–595.
3. van Zelst, A., & Schenk, T. C. W. (2004). Implementation of a MIMO OFDM-based wireless LAN system. *IEEE Transactions on Signal Processing, 52*(2), 483–494.
4. Wolniansky, P. W., Foschini, G. J., Golden, G. D., & Valenzuela, R. A. (1998). V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel. *Proc. ISSSE*, 295–300.
5. Golden, G. D., Foschini, C. J., Valenzuela, R. A., & Wolniansky, P. W. (1999). Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture. *Electronics Letters, 35*(1), 14–16.
6. Waters, D. W., & Barry, J. R. (2008). The Chase family of detection algorithms for multiple-input multiple-output channels. *IEEE Transactions on Signal Processing, 56*(2), 739–747.
7. Luo, Z., Zhao, M., Liu, S., & Lin, Y. (2008). Generalized parallel interference cancellation with near-optimal detection performance. *IEEE Transactions on Signal Processing, 56*(1), 304–312.
8. Hassibi, B. An efficient square-root algorithm for BLAST. http://mars.bell-labs.com/cm/ms/what/mars/index.html.
9. Benesty, J., & Huang, Y. (2003). A fast recursive algorithm for optimum sequential signal detection in a BLAST system. *IEEE Transactions on Signal Processing, 51*(7), 1722–1730.
10. Viterbo, E., & Boutros, J. (1999). A universal lattice decoder for fading channels. *IEEE Transactions on Information Theory, 45*(5), 1639–1642.
11. Damen, M. O., El Gamal, H., & Caire, G. (2003). On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions On Information Theory, 49*(10), 2389–2402.
12. Artés, H., Seethaler, D., & Hlawatsch, F. (2003). Efficient detection algorithms for MIMO channels: a geometrical approach to approximate ML detection. *IEEE Transactions on Signal Processing, 51*(11), 2808–2820.
13. Choi, W. J., Negi, R., & Cioffi, J. M. (2000). Combined ML and DFE decoding for the V-BLAST system. *Proceedings of the IEEE International Conference on Communications, 3*, 1243–1248.
14. Yang, L., Chen, M., Cheng, S., & Wang, H. (2004). Combined maximum likelihood and ordered successive interference cancellation grouped detection algorithm for multistream MIMO. *Proceedings of the IEEE International Symposium on Spread Spectrum Techniques and Application*, Aug.–Sep. 250–254.
15. Tarokh, V., Naguib, A., Seshadri, N., & Calderbank, A. R. (1999). Combined array processing and space-time coding. *IEEE Transactions on Information Theory, 45*(4), 1121–1128.
16. Al-Ghadhban, S., & Woerner, B. D. (2004). Iterative joint and interference nulling/cancellation decoding algorithms for multi-group space time trellis coded systems. *Proceedings IEEE Wireless Communications and Networking Conference (WCNC), 4*, 2317–2322.
17. Shen, C., Zhang, H., Dai, L., & Zhou, S. (2003). Detection algorithm improving V-BLAST performance over error propagation. *Electronic Letters, 39*(13), 1107–1108.
18. Li, D., Cai, L., & Yang, H. (2004). New iterative detection algorithm for V-BLAST. *IEEE Vehicular Technology Conference, 4*, 2444–2448.
19. Waters, D. W., & Barry, J. R. (2005). The sorted-QR chase detector for multiple-input multiple-output channels. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 1*, 538–543.
20. Li, Y., & Luo, Z. (2002). Parallel detection for V-BLAST system. *Proceedings of the IEEE International Conference on Communications, 1*, 340–344.
21. Lei, Z., Dai, Y., & Sun, S. (2005). A low complexity near ML V-BLAST algorithm. *Proceedings of the IEEE Vehicular Technology Conference (VTC), 2*, 942–946.
22. Wu, D. Y., & Van, L. D. (2008). A grouped-iterative framework for MIMO detection. *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Sep. 2008, accepted, Calgary, Canada.
23. Huang, C. J., Yu, C. W., & Ma, H. P. (2009). A power-efficient configurable low-complexity MIMO detector. *IEEE Transactions on Circuits and Systems, I, 56*(2), 485–496.
24. Wübben, D., Böhnke, R., Rinas, J., Kühn, V., & Kammeyer, K. (2001). Efficient algorithm for decoding layered space-time codes. *Electronic Letters, 37*(22), 1348–1350.
25. Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computations* (3rd ed.). Baltimore: Johns Hopkins University Press.
26. Burg, A., Borgmann, M., Wenk, M., Zellweger, M., Fichtner, W., & Bölcskei, H. (2005). VLSI implementation of MIMO detection using the sphere decoding algorithm. *IEEE Journal of Solid-State Circuits, 40*(7), 1566–1577.

Di-You Wu received the B.S. degree in mathematics from National Cheng Kung University, Tainan, Taiwan, in 2006 and the M.S. degree from Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, in 2008. He is working toward Ph.D.

degree at National Chiao Tung University. His research interests include VLSI digital signal processing and baseband communication systems.

Lan-Da Van received the B.S. (Honors) and the M.S. degree from Tatung Institute of Technology, Taipei, Taiwan, in 1995 and 1997, respectively, and the Ph. D. degree from National Taiwan University (NTU), Taipei, Taiwan, in 2001, all in electrical engineering.

From 2001 to 2006, he was an Associate Researcher at National Chip Implementation Center (CIC), Hsinchu, Taiwan. Since Feb. 2006, he joined the faculty of Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he is currently an Assistant Professor. His research interests are in VLSI algorithms, architectures, and chips for digital signal processing, 3D graphics, and baseband communication systems. This includes the design of high-performance /low-power /cost-effective 3D graphics processors, adaptive filters, transform, computer arithmetic, and platform-based system-on-a-chip (SOC) designs. He has published more than 40 journal and conference papers and held one US and one Taiwan patens in these areas.

Dr. Van was a recipient of the Chunghwa Picture Tube (CPT) and Motorola fellowships in 1996 and 1997, respectively. He was an elected chairman of *IEEE* NTU Student Branch in 2000. In 2002, he has received *IEEE* award for outstanding leadership and service to the *IEEE* NTU Student Branch. In 2005, he was a recipient of the Best Poster Award at iNEER Conference for Engineering Education and Research (*iCEER*). From 2009, he serves as the officer of IEEE Taipei Section. He served as a reviewer for the *IEEE TCAS I*, the *IEEE TCAS II*, the *IEEE TCSVT*, the *IEEE TC*, the *IEEE TMM*, the *IEEE TSP*, the *IEEE TVLSI Systems*, and the *IEEE SPL*. He is a member of the *IEEE*.