IET Communications

Research Article

IET Journals
The Institution of Engineering and Technology

Joint sparse graph over GF(q) for code division multiple access systems

ISSN 1751-8628
Received on 3rd July 2014
Accepted on 4th November 2014
doi: 10.1049/iet-com.2014.0616
www.ietdl.org

Lei Wen[1,2] ✉, *Man Su*[1]

[1]*Centre for Communication Systems Research, University of Surrey, GU2 7XH, UK*
[2]*College of Electronical Science and Engineering, National University of Defence Technology, 410073, People's Republic of China*
✉ *E-mail: l.wen@surrey.ac.uk*

Abstract: Low-density signature code division multiple access (LDS-CDMA) and low-density parity-check (LDPC) code can both be represented by a single sparse graph. In this study, the authors propose a joint sparse graph (JSG) over GF(q) which combines LDS-CDMA and non-binary LDPC codes, namely JSG-CDMA. On the JSG, multiple accessing and channel coding are well-linked together, in addition, joint detection and decoding are performed by message passing algorithm. Two schedules for message updating on the JSG, that is, flooding and serial schedules, are, respectively, presented. To predict the convergence behaviour of the joint detection and decoding, they depict the iterative structure of the JSG-CDMA receiver and analyse its extrinsic information transfer chart. Key factors of JSG-CDMA include message passing schedule, maximum iteration number and Galois field order. Simulation results show that JSG-CDMA performs much better than conventional CDMA, in addition, compared with LDS-CDMA and turbo structured LDS-CDMA, at a bit error rate of 3×10^{-4}, JSG-CDMA brings about 1.9 and 1.1 dB gain, respectively.

1 Introduction

Code division multiple access (CDMA) generates spreading code that runs at a much higher rate than the data to be transmitted, and has been applied in many communication standards [1]. In the uplink transmission of CDMA, the number of users naturally exceeds that of available chips, which results in an overloaded condition. In that case, the orthogonality of spreading signatures cannot be maintained [2–4], thus the multiuser interference (MUI) becomes inevitable even if complicated multiuser detection (MUD) such as combination of minimum-mean-square error (MMSE) and parallel interference canceller (PIC) is adopted [5, 6]. In order to deal with such problem, spreading codes under overload conditions have been designed [7–9], and low-density signature for un-coded CDMA (low-density signature code division multiple access – LDS-CDMA) has been proposed [10, 11]. The basic principles of the LDS-CDMA are (i) changing the interference pattern being seen by each user; and (ii) limiting the amount of interference occurred on each user. In LDS-CDMA systems, because of the low density structures, each data symbol is only spread over a limited number of chips (effective processing gain). Then, each user's generated chip is only used by a limited number of data symbols that may possibly belong to different users. Hence each user will experience interference from only a small number of other data symbols. Based on the low density structure, iterative message passing can be applied for a MUD with low complexity, and the MUI can be eliminated effectively.

Low-density signature of LDS-CDMA is very similar to the parity check matrix of low-density parity check (LDPC) codes which were discovered by Gallager in 1962 and shown to approach Shannon-limit in the late 1990s [12]. Recently, non-binary LDPC codes over Galois field (GF) have received considerable attention because of their excellent ability of error correction which is superior to that of RS codes and binary LDPC codes [13, 14]. Moreover, the impressive performance achieved by iterative decoding of turbo codes has encouraged researchers to consider applying this iterative architecture in multiuser transmissions, that is, MUD and channel decoding are considered to be linked by turbo style iterations [15]. Take the coded LDS-CDMA as an example, if the decoder output is fed back to the detector input through interleavers, it is referred to the turbo structured LDS-CDMA. By doing so, information on the detector and the decoder can be exchanged, and the system performance can be improved. However, the convergence behaviour of the turbo structured LDS-CDMA is not optimal. Therefore, for coded LDS-CDMA, designing a receiver to achieve satisfactory performance is a challenge. This paper aims to address this challenge by constructing a non-binary joint sparse graph (JSG) for CDMA (JSG-CDMA) systems. The contributions of this paper are as follows.

(1) Inspired by the similarity of LDS-CDMA and LDPC codes, we propose a JSG which combines LDS-CDMA and non-binary LDPC codes, namely JSG over GF(q) ($q > 2$) for CDMA (JSG-CDMA). Unlike any existing sparse graph that is only applied in one specific field, the JSG is novel as it combines multiple accessing (LDS-CDMA) and channel coding (non-binary LDPC codes) techniques.
(2) To the best of our knowledge, MUD and channel decoding has not been jointly performed on a whole sparse graph. Based on the message passing algorithm and the proposed JSG, we develop a joint detection and decoding algorithm for JSG-CDMA. Two strategies of message passing for the joint detection and decoding, that is, flooding and serial schedules, are, respectively, presented. It is noteworthy that the JSG-CDMA is significantly different from the LDS-CDMA and the turbo structured LDS-CDMA. For LDS-CDMA receiver, MUD and channel decoding are separately performed. For turbo structured LDS-CDMA receiver, detector and decoder are connected by turbo iterations. For JSG-CDMA receiver, however, there is a JSG rather than turbo structure. MUD and channel decoding can be simultaneously performed on the JSG.
(3) Iterative structures of the JSG-CDMA receiver are depicted, and extrinsic information transfer (EXIT) chart of the joint detection and decoding is analysed. Simulation results show that the JSG-CDMA outperforms similar well-known systems.

The rest of this paper is organised as follows. Section 2 presents the transmitter and receiver structures of JSG-CDMA. In Section 3, joint detection and decoding for JSG-CDMA is presented, including flooding and serial schedules. The iterative structures of JSG-CDMA and the EXIT chart are analysed in Section 4, and

then the simulation results and comparisons are given in Section 5. Finally, Section 6 is devoted to conclusions.

2 System model

Consider an uplink transmission and symbol-synchronous CDMA system with K users and processing gain of N. Each user has a data vector consisting of M modulated symbols. Let J be the parity check equations of non-binary LDPC code for each user. In CDMA, a chip refers to a pulse of a direct-sequence spread spectrum code, so there are N chips available for spreading in our scenario. We assume that perfect channel state information is available at the receiver, and all users are aligned in the phase. The diagrams of JSG-CDMA are shown in Fig. 1.

2.1 Transmitter model

As shown in Fig. 1*a*, the transmitter block diagrams provide a group of independent links for different users in addition to multiple-access channels. For each user, after non-binary LDPC encoding and symbol mapping, we multiply the modulated symbols with a spreading signature (a random sequence of chips) to perform the spreading process. Note that the LDPC code can be any code rate, which is depended on channel condition and performance requirement. In conventional CDMA, each user spreads its original data using a given spreading code (a sequence of N-chips), where each element of the spreading code takes, usually, non-zero values, which is optimised under certain criteria, for example, good auto- and/or cross-correlation properties. However, it is impossible for the spreading codes to obtain the orthogonality under overload conditions. Meanwhile, conventional spreading sequence naturally has high density, which means majority chips have non-zero values. Its drawback is that, each user will see the interference coming from all other users at the chip level. By contrast, in Fig. 1*a*, we can see that instead of optimising the N-chips sequences, the system intentionally arranges each user to spread its data over a small number of chips and then zero-padding is applied to keep the processing gain to be N. It is noteworthy that the zero-padding inserts zero elements in spreading sequences, thus the generated sequences have a maximum of $d_{v,\mathrm{lds}}$ non-zero values and N–$d_{v,\mathrm{lds}}$ zeros (the term non-zero means there is an edge connecting a chip and a data symbol, while the term zero means there is no edge between the chip and the data symbol), then they are interleaved uniquely for

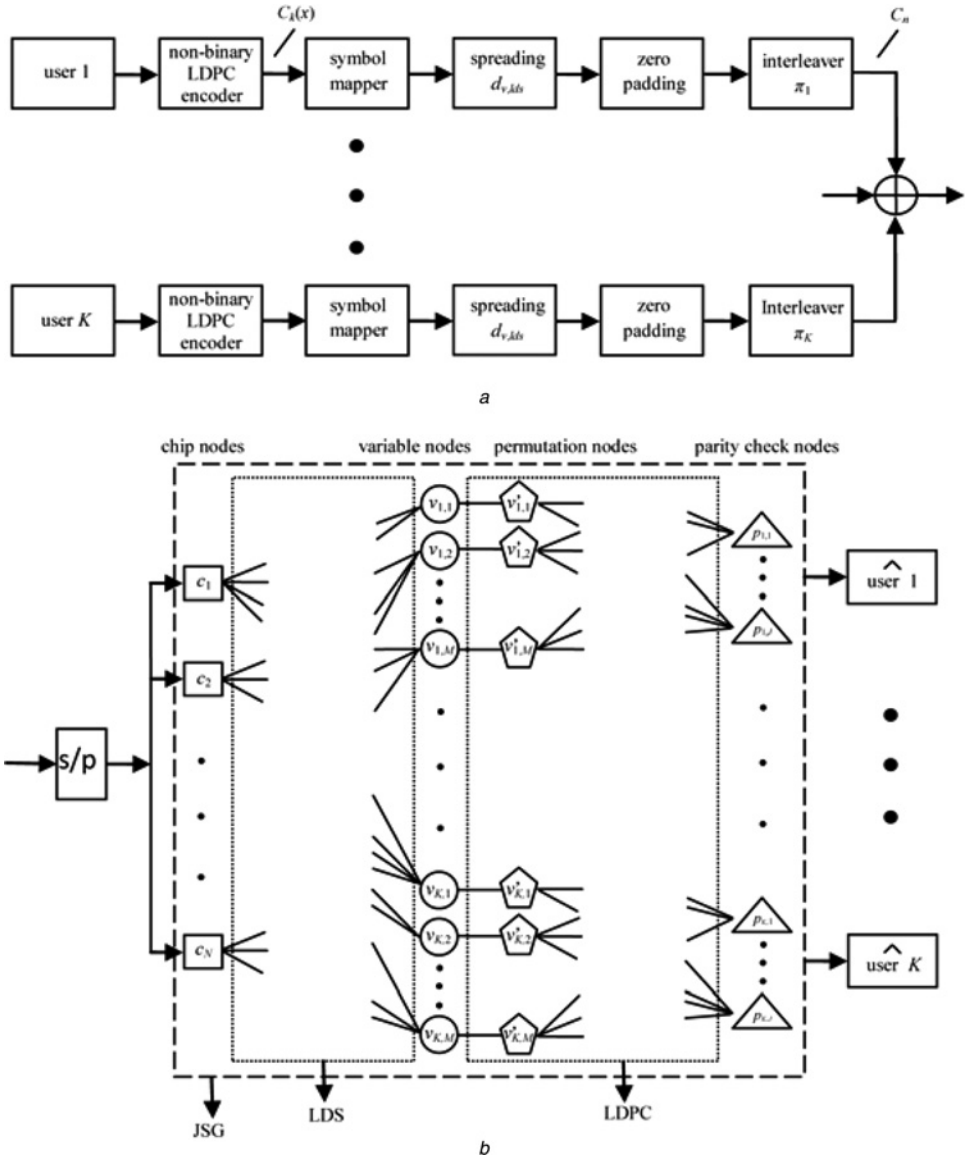

Fig. 1 *System model of JSG-CDMA*

a Transmitters
b Receiver

IET Commun., 2015, Vol. 9, Iss. 5, pp. 707–718

each user such that the resultant signature becomes very sparse. The interleaving process is designed so that at each received chip there exist a contribution of, instead of all users, only a small number of users. Consequently, the interference pattern being seen by each user is different. The generated spreading chips are denoted by C_n in Fig. 1*a*. Therefore interleavers are key functions in JSG-CDMA which control edge connections between chips and data symbols, and they are designed to fit certain parameters such as $d_{v,\mathrm{lds}}$ and cycle structure.

In transmitters of the JSG-CDMA, because of the low-density graph, each data symbol is only spread over $d_{v,\mathrm{lds}}$ chips. Each chip is only used by a limited number of data symbols that may belong to different users. Each user will experience interference from only a small number of other users data symbols. More explicitly, the number of symbols that are superimposed on each chip is much less than the total number of modulated symbols, and the number of chips that are spread by each symbol is much less than the total number of chips. In fact, $d_{v,\mathrm{lds}}$ is the effective spreading factor which determines the density of the spreading signature and the complexity of the joint detection and decoding. The $d_{c,\mathrm{lds}}$ to $d_{v,\mathrm{lds}}$ ratio is related to the system loading, where $d_{c,\mathrm{lds}}$ refers to the edge number connected to chip nodes. The size of the low-density signature depends on chips number and system loading.

2.2 Receiver model

As for the receiver in Fig. 1*b*, there are four types of nodes: chip nodes c_n ($n \in [1, N]$), parity check nodes $p_{k,j}$ ($k \in [1, K]$, $j \in [1, J]$), variable nodes $v_{k,m}$ ($k \in [1, K]$, $m \in [1, M]$) and permutation nodes $v'_{k,m}$ ($k \in [1, K]$, $m \in [1, M]$), representing the nth chip, the jth parity check equation of the kth user, the mth data symbol and its permutation value of the kth user, respectively. Owing to the non-binary values of the graph, the permutation nodes that connect the variable nodes and the parity check nodes perform a multiplication or a division over GF(q). For LDS-CDMA and LDPC code, it has been well studied to represent the de-spreading (removing the spread-spectrum code) and the parity check matrix by so-called bipartite graphs which are, respectively, labelled by LDS and LDPC in Fig. 1*b*. Note that the LDS and the LDPC both indicate single graph, and they, respectively, belong to multiple accessing and channel coding. In our proposal, variable nodes are used to connect parity check nodes (via permutation nodes) and chip nodes. Thus the receiver becomes a JSG over GF(q), which is labelled by JSG in the figure. As such, the LDS signature and the LDPC codes are perfectly linked together. The JSG is constructed according to the following rules: a chip node is connected to a variable node whenever such chip in the user's spreading sequence is non-zero; a parity check node is connected to a variable node via a permutation node whenever the corresponding value of the parity check equation is non-zero.

The proposed JSG is novel as it combines multiple access and sparse graph coding techniques. Note that the JSG is different from turbo processing principles, as there is no outer-inner turbo style iteration. In the following section, we will give an iterative algorithm of detection and decoding on the JSG.

3 Joint detection and decoding for JSG-CDMA

The algorithm iteratively computes the distributions of variables in a graph-based models and comes under different names, depending on the context. These names include: the sum–product algorithm (SPA), belief propagation algorithm (BPA), and message passing algorithm (MPA). The term 'message passing' usually refers to all such iterative algorithms, including the SPA (BPA) and its approximations [16–18]. Based on MPA, we give approaches for joint detection and decoding of JSG-CDMA.

The key points of a clear presentation of the joint detection and decoding are the use of a tensorial representation of the messages along the edges in the graph model, and transformations of the non-binary graph so that the MPA can be written in a simple way. In fact, the MPA can be presented as message update equations on the sparse graph, involving messages between different types of nodes via edges. In a typical run, each node of the graph calculates iteratively from the previous values of the neighbouring information (two nodes are said to be neighbours if they are connected by an edge). The information exchanged is the soft value that represents the reliability of the symbol related to an edge. In addition, the order of message passing between different nodes on graphs is referred as an updating schedule. We present a so-called flooding schedule for the joint detection and decoding in the sequel.

3.1 Flooding schedule

Without loss of generality, spreading signature for the kth user is represented by $\boldsymbol{S}_k = [s_{k,1,\ldots,}\ s_{k,\ M}] \in C^{N\times M}$, where C denotes the complex field. Let $\boldsymbol{S} = [\boldsymbol{S}_1, \ldots, \boldsymbol{S}_K] \in C^{N\times M\times K}$ and $\boldsymbol{H} = [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_K] \in C^{J\times M\times K}$ be the low-density spreading signatures for CDMA and the low-density parity check matrices for LDPC code, respectively. We also define $\boldsymbol{T} = \mathrm{diag}(\boldsymbol{T}_1, \ldots, \boldsymbol{T}_K)$ as the transmit power gain of users and $\boldsymbol{G}_k = \mathrm{diag}(g_{k,1}, \ldots, g_{k,N})$ as the corresponding channel gain for the kth user. Moreover, $\psi_n = \{(k, m): s^n_{k,m} \neq 0\}$ and $\varepsilon_{k,m} = \{n: s^n_{k,m} \neq 0\}$ are the set of data symbols (which may belong to different users) that interfere on chip c_n and the set of chips that $v_{k,\ m}$ is spread on, respectively; $\phi_j = \{(k, m): h^j_{k,m} \neq 0\}$ and $\omega_{k,m} = \{j: h^j_{k,m} \neq 0\}$ are the set of permutation nodes that connect to parity check node $p_{k,j}$ and the set of parity check nodes that connect to $v'_{k,m}$, respectively. The MPA on the JSG is not a direct generalisation of the binary case because the elements of the graph are non-binary values. A parity check node in Fig. 1*b* represents a parity check equation, that is

$$\sum_{k=1}^{d_{p,\mathrm{ldpc}}} h_k(x)c_k(x) = 0 \mod p(x) \tag{1}$$

where $p(x)$ in the modulo operator is a degree p–1 primitive polynomial of GF(q), $c_k(x)$ is the codeword symbols and $d_{p,\mathrm{ldpc}}$ is the maximum number of edges connected to the parity check nodes. This equation shows that the variable nodes needed to perform the MPA are not the codeword symbols alone, but the codeword symbols multiplied by non-zero values of the parity check matrices $\boldsymbol{H}$. The transformation of the JSG can be performed by adding permutation nodes which correspond to the multiplication or division of the non-zero values.

In the receiver of JSG-CDMA, the received spreading sequence for the data symbol m of the kth user can be represented by $\boldsymbol{r}_{k,m} = \boldsymbol{T}_k\boldsymbol{G}_k s_{k,m}$. In particular, the received signature gain at the nth chip of the variable node $v_{k,m}$ is $r^n_{k,m} = \mathbf{T}_k g_{k,n} s^n_{k,m}$. For the uplink transmission, the received signal corresponding to the nth chip can be written as

$$y_n = \sum_{k=1}^{K}\sum_{m=1}^{M} r^n_{k,m}v_{k,m} + z_n \tag{2}$$

where z_n is additive white Gaussian noise (AWGN) with variance σ^2_A and mean zero. Considering that in the low-density graph, the signature has a limited number of non-zero values, we can express the received signal at the nth chip as

$$y_n = \sum_{(k,m)\in\psi_n} r^n_{k,m}v_{k,m} + z_n \tag{3}$$

The log-likelihood ratio (LLR) exchanged between different kinds of nodes in Fig. 1*b* is denoted by the upper case L. The flooding schedule for the joint detection and decoding is presented in Appendix 1. Message updating of different types of nodes is depicted in Fig. 2. It can be seen that in Fig. 2*a*, the messages from the channel and the extrinsic LLR from the connected variable nodes $v_{1,2}$ and $v_{2,3}$, are fed into the chip node c_1 to compute $L_{c_1\to v_{3,1}}$. In Fig. 2*b*, the extrinsic messages from the

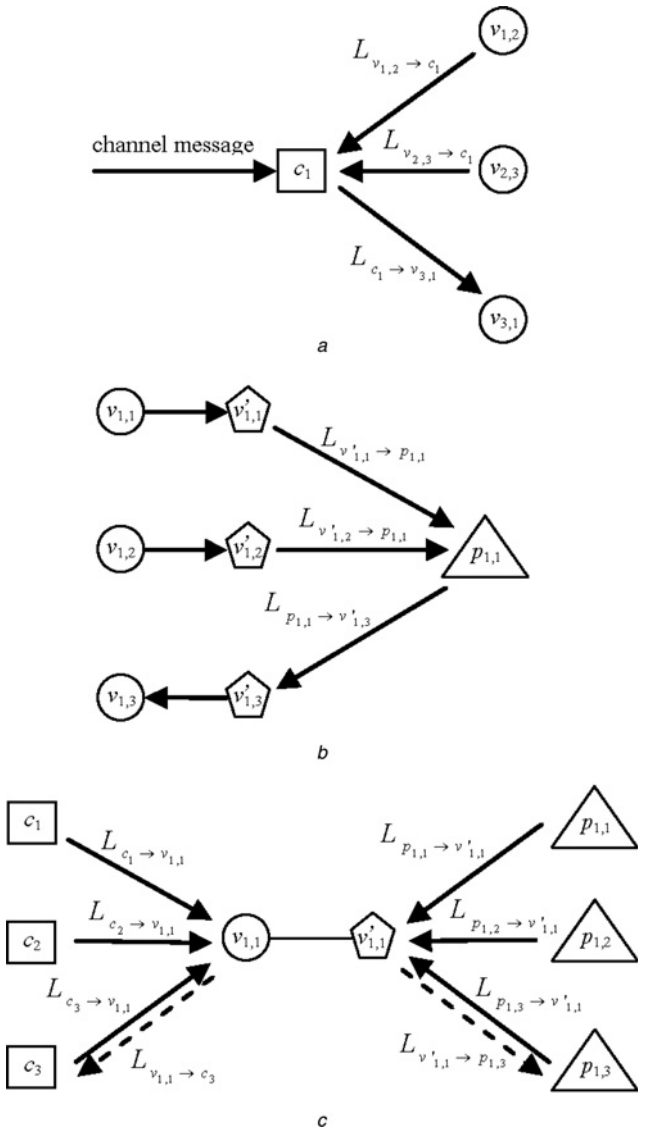

Fig. 2 *Message updating for different types of nodes in JSG-CDMA*

a Chip nodes updating
b Parity check nodes updating
c Variable nodes and permutation nodes updating

variable nodes, $v_{1,1}$ and $v_{2,1}$, are, respectively, delivered to the permutation nodes, $v'_{1,1}$ and $v'_{2,1}$, then the permutated messages are fed into the parity check node $p_{1,1}$ to compute $L_{p_{1,1}\to v'_{3,1}}$. In Fig. 2*c*, extrinsic messages from both chip nodes and parity check nodes (marked by solid lines) are delivered into the variable node $v_{1,1}$ or the permutation node $v'_{1,1}$ to compute $L_{v_{1,1}\to c_3}$ and $L_{v'_{1,1}\to p_{1,3}}$ (marked by long dashed lines). As a result, detection and decoding are well combined in the JSG.

3.2 Serial schedule

Message passing schedule not only influences the convergence rate, but also affects the system performance. In a cycle-free graph, the belief will converge to the exact a posterior probability after a finite number of iterations that is bounded by a half-length of the longest path of the graph. Nevertheless, cycles cannot be avoided, and the propagated information may lead to inaccurate a posterior probability. In the flooding schedule presented above, messages are updated in a parallel manner, that is, all chip nodes and parity check nodes update at the same time, and then all variable nodes update simultaneously. In a typical run, the updated message has to be buffered until current iteration terminates, which means the new message cannot join the belief propagation immediately. The drawback of the flooding schedule is that the convergence behaviour of message passing is not ideal. In order to accelerate the convergence rate, we propose a serial schedule for the message updating on the JSG.

In the serial schedule, chip nodes update message sequentially. We use $L^i_{v_{k,m}}$ and $L^i_{c_n\to v_{k,m}}$ to compute $L^i_{v_{k,m}\to c_n}$ on the fly, avoiding additional memory to store $L^i_{v_{k,m}\to c_n}$, where the superscript i refers to the ith iteration. Such processing can be derived by combining (9) and (25) as

$$L^i_{v_{k,m}} = L^{i-1}_{v_{k,m}} - L^{i-1}_{c_n\to v_{k,m}} + L^i_{c_n\to v_{k,m}} \tag{5}$$

Detailed procedures of the serial schedule are presented in Appendix 2. Apparently, in the flooding schedule, new messages can only be used in the next iteration. In contrast, the serial schedule allows immediate propagation of new messages, thus it is more efficient in message convergence. As for hardware implementation, the flooding schedule has inherent advantages, namely easier parallelisation of the algorithm for high speed. Therefore, in practical system design, the choice of message passing schedule depends on specific requirements, that is, receiver complexity and latency.

4 Exit chart analysis for JSG-CDMA

An extrinsic information transfer chart, commonly named an EXIT chart, is a technique to aid the construction of good iteratively decoded forward-error-correcting codes (in particular LDPC codes and Turbo codes) [19]. An EXIT chart includes the response of elements of an iterative decoder, where the response can either be seen as extrinsic information or a representation of the messages in belief propagation. This observation is supported by a large number of iterations for information exchanging when decoding happens. However, the EXIT chart has not been applied to analyse a JSG. Based on the proposed JSG-CDMA system model, we use the EXIT chart to analyse the convergence behaviour of the joint detection and decoding.

4.1 Iterative structure of JSG

The philosophy of the JSG in JSG-CDMA is that if a fraction of signal of some users is superimposed by a fraction of signals coming from a relatively small number of interferers, then the search-space should be smaller, consequently, detection and decoding techniques with affordable complexity can be used to recover the corrupted part of the signal. In addition, apart from being practical for implementation, the graph model with low density also is benefited from having the intrinsic interference diversity by avoiding strong interferers to corrupt all chips of a user. Fig. 3*a* depicts the matrix representation of the JSG, where the α represents non-zero value randomly chosen from GF(q). In this figure, the upper rows of the matrix correspond to the parity check nodes of different users, while the lower rows correspond to the chip nodes. The columns of the matrix represent the variable nodes or the permutation nodes which belong to different users. Apparently, each user has an independent parity check matrix for forward error correction, while all parity check matrices are linked to each other through the spreading matrix. Therefore the parity check matrices and the spreading matrix form a joint sparse matrix. The cycle is an important factor in graph models, as short cycles may lead to failure of message convergence or oscillation between multiple states over repeat iterations. Usually,

$$L^i_{c_n\to v_{k,m}} = \kappa_{n,k,m} \max^{*}_{\mathbf{v}_{[n]}} \left(\sum_{(k',m')\in\psi_n\setminus(k,m)} \left(L^{i-1}_{v_{k',m'}} - L^{i-1}_{c_n\to v_{k',m'}} \right) - \frac{1}{2\sigma_A^2} ||y_n - \mathbf{r}^{\mathrm{T}}_{[n]}\mathbf{v}_{[n]}||^2 \right) \tag{4}$$

IET Commun., 2015, Vol. 9, Iss. 5, pp. 707–718

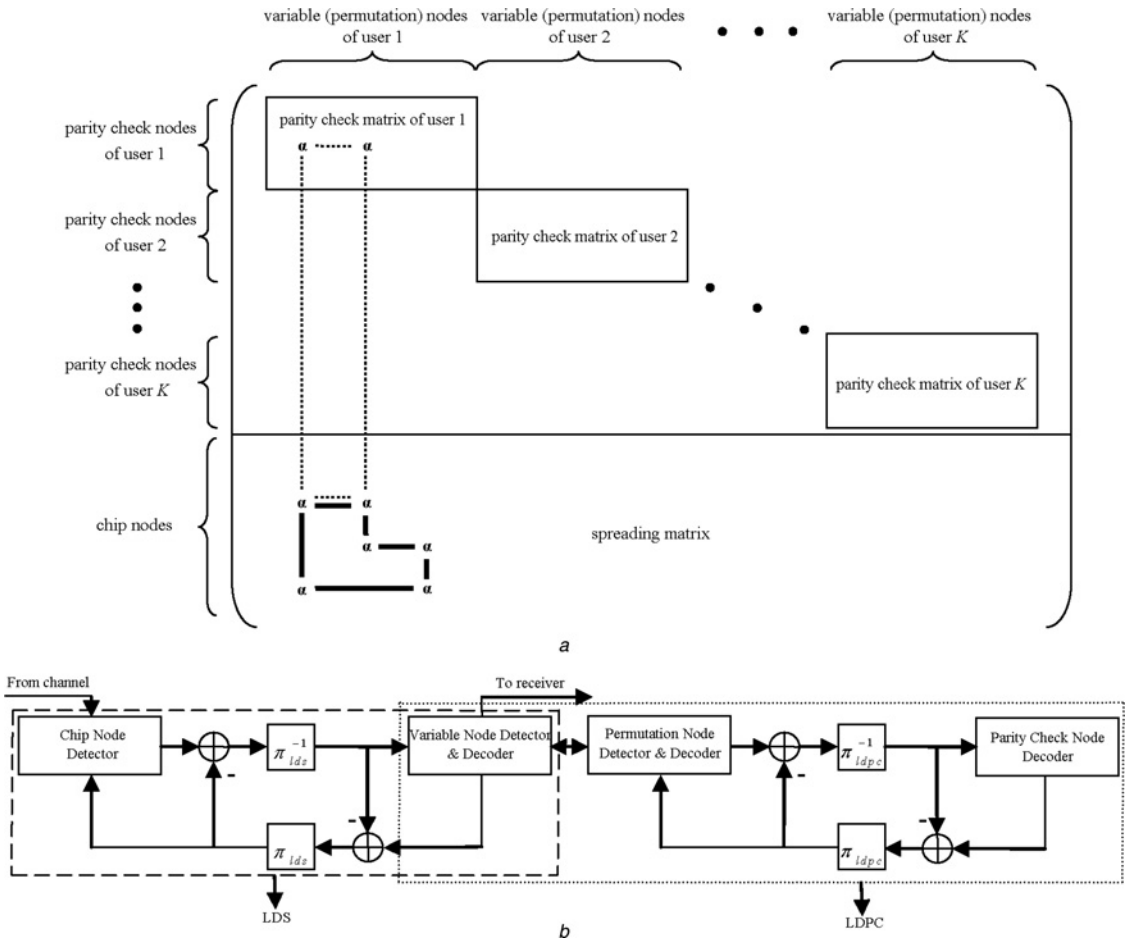

Fig. 3 *Iterative structure of the JSG*

a Matrix form of the JSG
b Components of the JSG

short cycles are only considered and avoided in a single matrix. For example, in the spreading matrix, length-4 cycles are deleted, thus the shortest cycle length of the spreading matrix is 6, which is marked by bold lines in the figure. Nevertheless, in the joint matrix, length-4 cycles are easy to be regenerated without careful design (marked by dashed lines in the figure). Short cycles, especially length-4 cycles, degrade the performance of message passing on a graph, thus it is necessary to remove length-4 cycles in the joint matrix rather than any single matrix.

According to Fig. 3*a*, variable nodes calculate the extrinsic messages of chip nodes using a priori information which they receive from other connected chip nodes and parity check nodes. Meanwhile, based on the received a priori information, permutation nodes calculate the extrinsic messages of parity check nodes. The same rule can be applied to the extrinsic messages that the chip nodes and parity check nodes send to variable nodes or permutation nodes. To evaluate the transformation of extrinsic information on the JSG, the sets of chip nodes, variable nodes, permutation nodes and parity check nodes are referred to as a chip nodes detector (CND), variable node detector–decoder (VNDD), permutation node detector–decoder (PNDD) and parity check node decoder (PND), respectively. Fig. 3*b* shows the structure of the iterative detector and decoder for JSG-CDMA. As depicted in the figure, the extrinsic LLR that has been passed on are considered as a priori information by the other detector or decoder. The edge interleavers connect different types of nodes, each of which represents a sparse signature or a sparse matrix. Note that such iterative structure is more complicated than any previous single graph which only has the long dash box (LDS-CDMA) or the dash box (LDPC code) in Fig. 3*b*.

4.2 Exit chart analysis

4.2.1 Exit curve for VNDD and PNDD: In this paper, $I_{\text{A, VNDD and PNDD}}$ refers to the average mutual information between the bits on the VNDD and PNDD edges and the a priori LLR, $I_{\text{E, VNDD and PNDD}}$ is the average mutual information between the bits on the VNDD and PNDD edges and the extrinsic LLR. In order to compute an EXIT curve for variable nodes, $L_{c_n \to v_{k,m}}$ and $L_{p_{k,j} \to v'_{k,m}}$ are modelled as the soft output of an AWGN channel. Then the mutual information between the variable node's extrinsic messages and actual values of symbols on the edges is calculated. A priori LLR can be calculated by

$$A = \mu_A x + z_n \tag{6}$$

where z_n is AWGN with variance σ_A^2 and mean zero; $x \in \pm 1$ is the bits on the graph edge. Furthermore

$$\mu_A = \frac{\sigma_A^2}{2} \tag{7}$$

The mutual information $I_{A,\text{ VNDD and PNDD}} = I(X;A)$ can be calculated by

$$I_{A,\text{VNDD and PNDD}} = \frac{1}{2}\sum_{x=-1,1}\int_{-\infty}^{+\infty} p_A(\beta|X=x)\log_2\frac{2p_A(\beta|X=x)}{p_A(\beta|X=-1)+p_A(\beta|X=1)}\,\mathrm{d}\beta \tag{8}$$

Since the conditional probability density function $p_A(\beta/X=x)$ depends on the LLR of A, we can write

$$I_{A,\text{VNDD and PNDD}}(\sigma_A) = 1 - \int_{-\infty}^{+\infty} \frac{e^{-((\beta-\sigma_A^2/2)^2/2\sigma_A^2)}}{\sqrt{2\pi}\sigma_A} \log_2(1+e^{-\beta})\,d\beta \tag{9}$$

For abbreviation, we define

$$B(\sigma) := I_{A,\text{VNDD and PNDD}}(\sigma_A = \sigma) \tag{10}$$

with

$$\lim_{\sigma\to 0} B(\sigma) = 0 \tag{11}$$

$$\lim_{\sigma\to \infty} B(\sigma) = 1 \tag{12}$$

where $\sigma \geq 0$. Considering (25) and (27) together with the fact that the sum of two normally distributed random variables is also normally distributed with the mean and variance equal to the sum of theirs, the EXIT function of a variable node can be expressed as

$$I_{E,\text{VNDD and PNDD}}(I_{A,\text{VNDD and PNDD}}, d_{v,\text{lds}}, d_{v,\text{ldpc}}) = B\left(\sqrt{(d_{v,\text{lds}} + d_{v,\text{ldpc}} - 1)(B^{-1}(I_{A,\text{VNDD and PNDD}}))^2}\right) \tag{13}$$

where $d_{v,\text{lds}}$ and $d_{v,\text{ldpc}}$ are the degrees of variable node in sparse graphs of LDS and LDPC, respectively.

Fig. 4*a* plots EXIT curves for different VNDD and PNDD. We assume that the message to and from an element in Fig. 3*b* can be described by a single number, that is, the extrinsic information. This is true when the messages are samples from a Gaussian distribution with the correct extrinsic information. The other assumption is that the messages are independent. According to (45) and Fig. 4*a*, unlike any single graph where only one type of node is considered, both LDS and LDPC nodes affect the VNDD and PNDD performance of the JSG.

4.3 Exit curve for CND and PND

Let $I_{A,\text{ CND and PND}}$ refer to the average mutual information between the bits on the CND and PND edges and the a priori LLR, $I_{E,\text{ CND and PND}}$ is the average mutual information between the bits on the CND and PND edges and the extrinsic LLR. A chip node has incoming messages from the connected variable nodes and the channel, whereas a parity check node only has messages coming from neighboured permutation nodes. The output LLR of chip nodes and parity check nodes are calculated by (9) and (24), respectively. We model $L_{v_{k,m}\to c_n}$ and $L_{v_{k,m}\to p_{k,j}}$ as the output of the channel that the input is the corresponding transmitted bit, and then calculate the mutual information of the output with regards to the actual value on the edges. Owing to the complexity of the calculation in chip nodes and parity check nodes, their EXIT curves are computed by simulations over channels. The probability density function for extrinsic information is determined by Monte Carlo simulation with histogram measurements, the mutual information between the extrinsic information and the bits on the joint graph edges, is subsequently calculated.

Fig. 4*b* shows the EXIT chart for JSG with different message passing schedules, where the channel model is Rayleigh fading channel at $E_b/N_0 = 10$ dB. Note that the EXIT chart technique is not limited to AWGN channel, it can also be applied to multipath fading channels when perfect channel state information is available at the receiver. The E_b/N_0 refers to the mean energy per bit to noise power spectral density ratio. The system parameters include: $d_{v,\text{lds}} + d_{v,\text{ldpc}} = 5$, the JSG is over GF($2^2$), user number is 96, chip number is 64 and system loading is 150%. In Fig. 4*b*, the behaviours of the detector and decoder are plotted on a two-dimensional chart. One component is plotted with its input on the horizontal axis and its output on the vertical axis. The other component is plotted with its input on the vertical axis and its output on the horizontal axis. The joint detection and decoding paths followed is found by stepping between the two curves. For a successful detection and decoding, there must be a clear swath between the curves so that iterative message passing can proceed from 0 bits of extrinsic information to 1 bit of extrinsic information. According to the figure, we can summarise:

(a) For flooding and serial schedules, intersection points of VNDD and PNDD and CND and PND almost overlap, indicating that the final convergence point and system performance will be the same for these two schedules.
(b) The trajectories of joint detection and decoding are also plotted in the figure. In the flooding schedule, six iterations are needed to reach the intersection point, while the serial schedule only occupies four iterations to the same position. These trajectories indicate that compared with the flooding schedule, the serial schedule is able to utilise more fresh and reliable information during the iterative process, hence its convergence rate is accelerated and improved. In other words, in some applications where there is a constraint on the number of allowable or affordable iterations because of the hardware cost or other specific reasons, the serial schedule can achieve much better performance than the flooding schedule thanks to the faster convergence rate, and the receiver complexity can be reduced because of the less iterations. It will be further confirmed in the simulation section.

5 Simulation results

In this section, the JSG-CDMA is simulated and analysed in terms of convergence rate, comparison with existing techniques, effect of GF order and near-far problem. We still choose $d_{v,\text{lds}} + d_{v,\text{ldpc}} = 5$ and Rayleigh fading channels for the following simulations.

5.1 Convergence rate

To verify the convergence rate of different message passing schedules, we show the performance at each iteration for the JSG-CDMA over GF(2^2) in Fig. 5, where the chip number is 64 and the system loading is 150%. As can be seen that in the first iteration, the flooding and the serial schedules start at the same point since there is no a priori probability available in the beginning, which have been shown in (4) and (32), consequently the updated messages are the same at this stage for both schedules. As iterations go on, the bit error rate (BER) drops dramatically. During the medium iterations, that is, from iteration of 2–4, the serial schedule attains much lower BER than the flooding schedule. For instance, in the third iteration, when E_b/N_0 equals to 10 and 16 dB, the BER in the flooding schedule are 3.0×10^{-2} and 1.2×10^{-2}, respectively, while the serial schedule can, respectively, achieve 1.9×10^{-2} and 3.0×10^{-3}. Obviously at this stage, the BER gap between these two schedules becomes wider as the increase of E_b/N_0. Furthermore, for the flooding schedule, the BER stops falling down after 5 or 6 iterations, which has been accurately predicted by the joint detection and decoding trajectory presented in Fig. 4*b*. In terms of the serial schedule, only four iterations are needed for message convergence, which also concurs with the joint detection and decoding trajectory shown in Fig. 4*b*. Therefore, EXIT chart analysis is confirmed by BER simulations, and more importantly, the serial schedule shows much faster convergence speed than that of the flooding schedule.

5.2 Comparison with existing techniques

In this subsection, the performance of JSG-CDMA is evaluated and compared with state-of-the-art techniques such as conventional CDMA, LDS-CDMA and turbo structured LDS-CDMA. For fair comparisons, a half rate LDPC code over GF(2^2) is applied to all

IET Commun., 2015, Vol. 9, Iss. 5, pp. 707–718

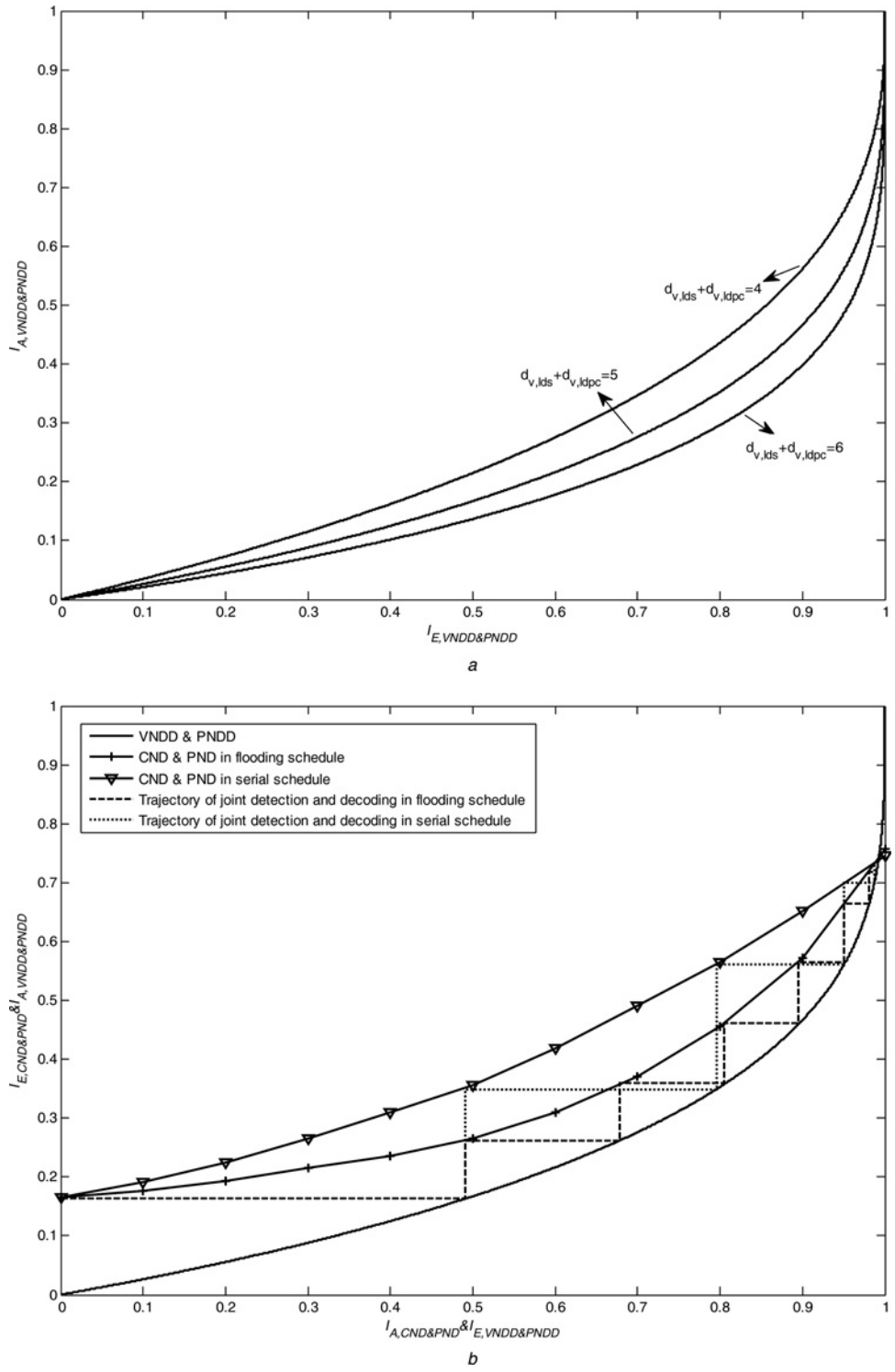

Fig. 4 *EXIT chart analysis for JSG-CDMA*

a EXIT curves of VNDD and PNDD
b EXIT chart for JSG over Rayleigh fading channels

the investigated systems. The chip number is 64 and the system loading is 150%. For CDMA system, Welch bound equality that minimises the variance of the MUI, is used for the 150% loaded sequences [20], and a MMSE-based PIC detector is used for MUD. The MMSE detector will give an unconstrained MMSE estimate of the transmitted symbols and PIC will smooth it further with the smoothing-coefficient varies in ascending-order from 0.6 to 0.8 with the same step size. For LDS-CDMA, the effective spreading factor is 3 and the MUD being used is an iterative detector with maximum iteration of 6 [7]. For turbo structured LDS-CDMA, there are 6 turbo style iterations between the detector and the decoder. For JSG-CDMA, flooding and serial schedules are, respectively, applied for the joint detection and decoding. Fig. 6 shows performance comparisons between these systems, and several conclusions can be made in the following:

(1) The conventional CDMA (labelled by CDMA MMSE-PIC) is inferior to the other systems, as its MMSE-PIC detector fails to attain a satisfactory performance under the overloaded condition.

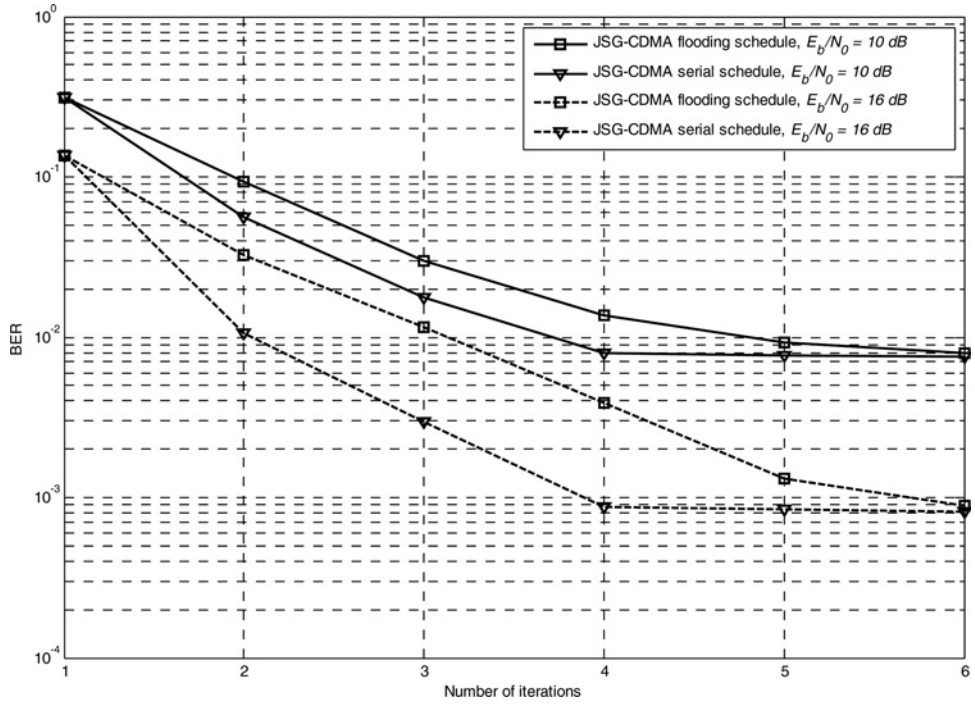

Fig. 5 *Performance at different iterations for JSG-CDMA*

(2) The LDS-CDMA outperforms the JSG-CDMA with three iterations (including flooding and serial schedules), but cannot achieve the performance of the JSG-CDMA with six iterations (including flooding and serial schedules).
(3) Compared with LDS-CDMA, the turbo structured LDS-CDMA brings about 1 dB gain in the high SNR region. This is related to the information exchange between the detector and the decoder in turbo iteration. However, the turbo structured LDS-CDMA is inferior to the JSG-CDMA with six iterations (including flooding and serial schedules), as the message passing on the JSG is able to fully utilise iterative information.
(4) For the JSG-CDMA with three iterations, the serial schedule shows much better performance than the flooding schedule. Take the BER of 3×10^{-3} as an example, compared with the flooding schedule, the serial schedule gains more than 6 dB. Moreover, as the E_b/N_0 increasing, the gap between these two schedules becomes wider, which has been indicated in Fig. 5. In fact, the performance of serial schedule with three iterations in JSG-CDMA is close to that of LDS-CDMA with six iterations.
(5) For the JSG-CDMA with six iterations, the curves of the flooding and the serial schedules nearly overlap, indicating that both schedules will converge to the same point and the BER results are consequently

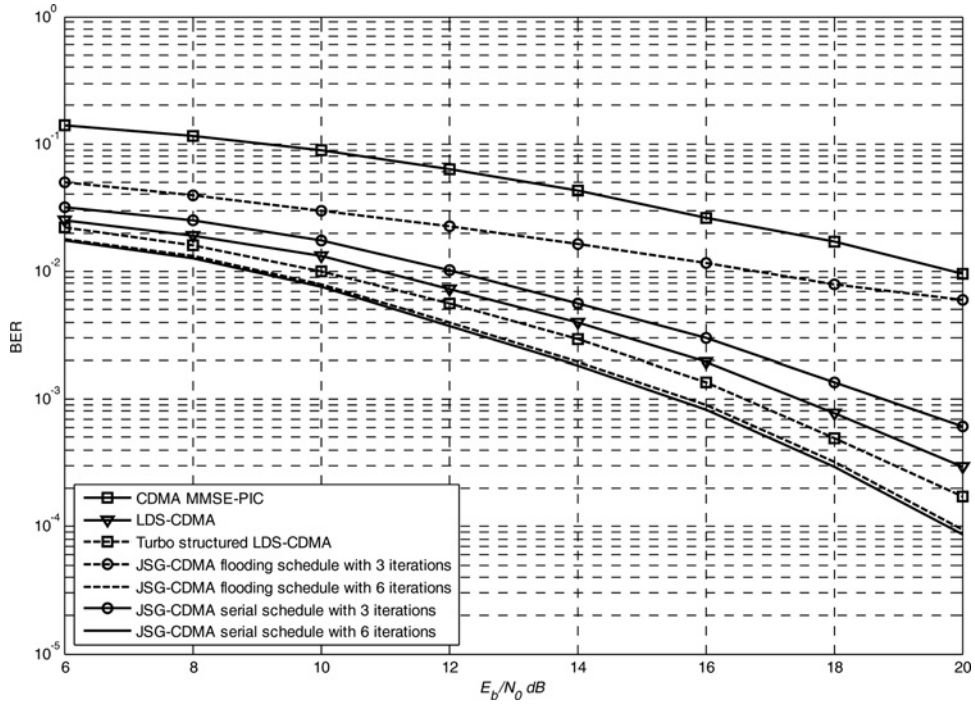

Fig. 6 *Performance comparisons between different systems*

IET Commun., 2015, Vol. 9, Iss. 5, pp. 707–718

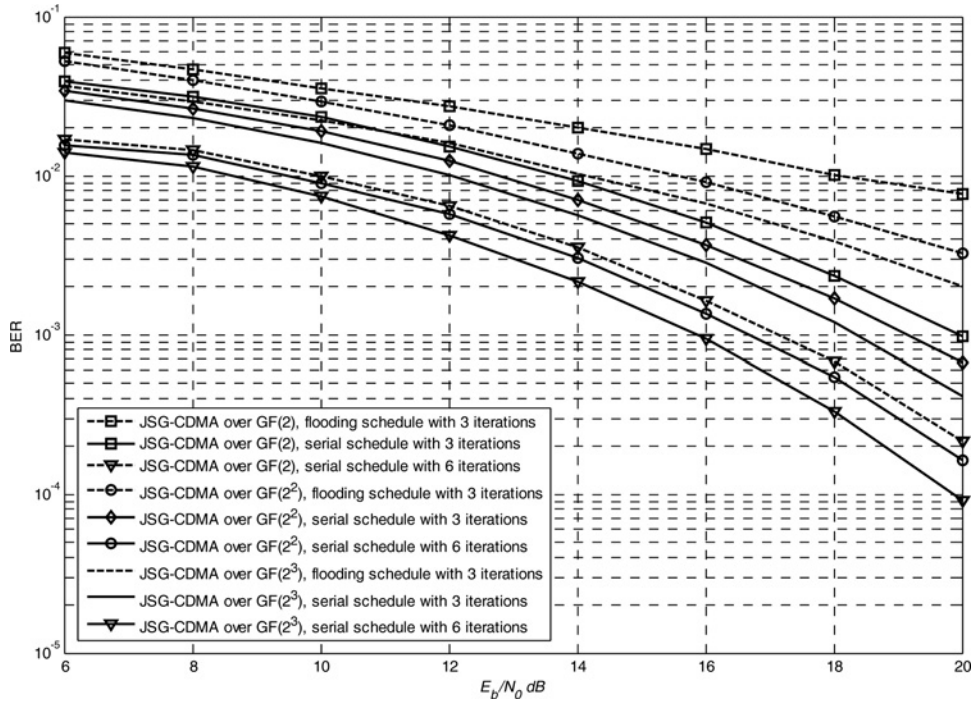

Fig. 7 *Effect of GF order*

the same. The results correspond to the EXIT chart presented in Fig. 4*b*, and also confirm the accuracy of the EXIT chart analysis. At BER of 3×10^{-4}, JSG-CDMA outperforms LDS-CDMA and turbo structured LDS-CDMA about 1.9 and 1.1 dB, respectively.

Considering all the results, it can be summarised that the JSG-CDMA performs much better than the conventional CDMA, and outperforms LDS-CDMA and turbo structured LDS-CDMA. In terms of the message passing schedule in JSG-CDMA, the serial schedule accelerates the convergence rate and can obtain a satisfactory performance even when limited iterations, that is, three iterations, are carried out.

5.3 Effect of GF order

To gain more insight to the JSG-CDMA, we present another result on the effect of GF order. Fig. 7 shows the performance of the JSG-CDMA over GF(2), GF(2^2) and GF(2^3), that is, different GF order. In fact, the GF(2) is the binary case of a JSG. The chip number is 128 and the system loading is 200% for all the scenarios. According to the figure, for each GF order, when three iterations are performed, the serial schedule always outperforms the flooding schedule. It is noteworthy that the performance can be improved when higher GF order is adopted. More explicitly, at a BER of 3.0×10^{-4}, GF(2^3) brings about 0.7 and 1.5 dB gain over GF(2^2) and GF(2), respectively. Therefore, GF order is an

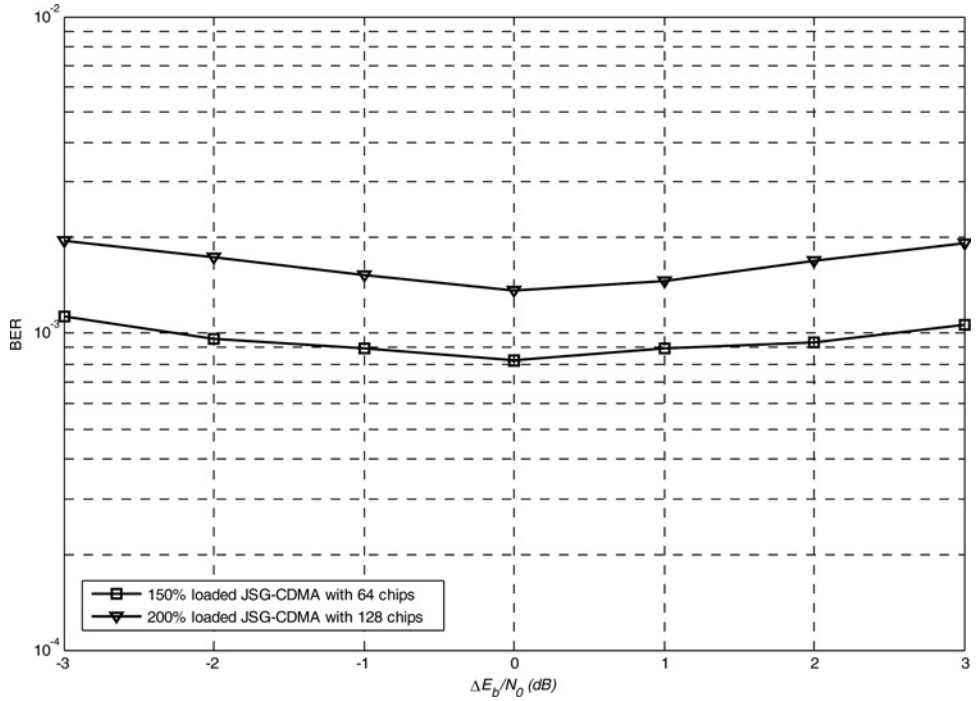

Fig. 8 *Near-far effect in JSG-CDMA*

important factor in the system performance. Generally speaking, higher order is able to achieve better performance, although its calculation complexity also becomes higher. In practical system design, we can choose appropriate GF order according to the available hardware sources.

5.4 Near-far problem

The joint detection and decoding presented in Section 3 is one of the receiver technologies for detecting desired signals from the interference and the noise. Traditionally, single-user receiver is known to suffer from the near-far problem, where a near-by or the strong signal source may block the signal reception of a far-away or weak signalled user. The near-far problem is more serious in CDMA-type wireless multi-user communication systems. Hence, it is necessary to investigate the near-far effect of the JSG-CDMA. Fig. 8 shows the performance of near-far resistance for the JSG-CDMA with different loadings and chips over GF(2^2). The simulation is carried out for the case when $E_b/N_0 = 16$ dB for the first user, and E_b/N_0 of other users is different. The BER performance of the first user is plotted against $\Delta E_b/N_0$, which represents the difference in E_b/N_0 between the user of interest and the other users. It can be seen that unequal received power has a minor effect on the performance of user of interest under different loading conditions. It is because of the iterative joint processing being employed in the receiver algorithm, or in other words the near-far problem can be alleviated by the JSG and the effective MPA. Therefore the power control does not need to be perfect for JSG-CDMA. In order to achieve an even better near-far resistance, the JSG-CDMA can resort to sophisticated power control mechanisms if necessary.

6 Conclusion

In this paper, a JSG over GF(q) for CDMA systems was introduced and analysed. Unlike any existing sparse graphs, the JSG is novel as it combines multiple accessing (LDS-CDMA) and channel coding (non-binary LDPC codes) techniques, and its receiver is different from turbo structured receivers. Based on the MPA, MUD and channel decoding can be performed simultaneously on the JSG. Different schedules for the joint detection and decoding, that is, flooding and serial schedules, are, respectively, presented. The flooding schedule has advantage of high speed implementation for parallel processing, while the serial schedule can obtain satisfactory performance in very limited iterations. In practical system design, it is necessary to strike a balance between hardware cost and system latency, and choose an appropriate schedule. According to the iterative structures of the JSG-CDMA receiver, EXIT charts and trajectories of joint detection and decoding are analysed, and the convergence behaviour of different schedules is verified by BER evaluation. Numerical results illustrate that the JSG-CDMA outperforms similar well-known systems. It was shown that JSG-CDMA performs much better than the conventional CDMA, in addition, compared with LDS-CDMA and turbo structured LDS-CDMA, at BER of 3×10^{-4}, JSG-CDMA, respectively, brings about 1.9 and 1.1 dB gain. In terms of GF of the JSG, higher order represents better performance, although its calculation complexity is increased. At BER of 3.0×10^{-4}, GF(2^3) brings about 0.7 and 1.5 dB gain over GF(2^2) and GF(2), respectively. Therefore, the performance of JSG-CDMA is determined by the message passing schedule, maximum iteration number and GF order. For high order modulation, that is, quadrature amplitude modulation or q-orthogonal modulation sets with large q, the non-orthogonal signature and coherent detection may result in imperfect performance. In that case, we have to take advantage of sparse code multiple access [21], that is, multiplex layer signatures, to design a more advanced JSG, which needs further detailed research.

7 Acknowledgments

The funding leading to this work is from the 5G innovation centre of University of Surrey and the United Kingdom Engineering and Physical Science Research Council (EPSRC) under grant number EP/J017655/1.

8 References

1 Daoben, L.: 'The perspectives of large area synchronous CDMA technology for the fourth-generation mobile radio', *IEEE Commun. Mag.*, 2003, **41**, (3), pp. 114–118
2 Dinan, E.H., Jabbari, B.: 'Spreading codes for direct sequence CDMA and wideband CDMA cellular networks', *IEEE Commun. Mag.*, 1998, **36**, (9), pp. 48–54
3 Hsiao-Hwa, C., Hsin-Wei, C., Guizani, M.: 'Orthogonal complementary codes for interference free CDMA technologies', *IEEE Wirel. Commun.*, 2006, **13**, (1), pp. 68–79
4 Bacci, G., Luise, M.: 'Performance comparison of energy-efficient power control for CDMA code acquisition and detection'. IEEE Global Telecommunications Conf., 2010, pp. 1–5
5 Al-Kamali, F.S., Dessouky, M.I., Sallam, B.M.: 'Regularized multi-stage parallel interference cancellation for downlink CDMA systems', *IET Commun.*, 2009, **3**, (2), pp. 239–248
6 Chi-Hsiao, Y.: 'MMSE multiuser receiver for uniformly quantized synchronous CDMA signals in AWGN channels', *IEEE Commun. Lett.*, 2007, **11**, (9), pp. 705–707
7 Khachatrian, G.H., Martirossian, S.S.: 'A new approach to the design of codes for synchronous-CDMA systems', *IEEE Trans. Inf. Theory*, 1995, **41**, (5), pp. 1503–1506
8 Ying-Wah, W., Hang, S.-C.: 'Coding scheme for synchronous-CDMA systems', *IEEE Trans. Inf. Theory*, 1997, **43**, (3), pp. 1065–1067
9 Ying-Wah, W., Chang, S.-C.: 'Coding scheme for noiseless and noisy asynchronous CDMA systems', *IEEE Trans. Inf. Theory*, 2010, **56**, (8), pp. 3786–3792
10 Wathan, F.P., Hoshyar, R., Tafazolli, R.: 'Iterated SISO MUD for synchronous un-coded CDMA systems over AWGN channel: performance evaluation in overloaded condition'. Int. Symp. on Communications and Information Technologies, 2007, pp. 397–402
11 Hoshyar, R., Wathan, F.P., Tafazolli, R.: 'Novel low-density signature for synchronous CDMA systems over AWGN channel', *IEEE Trans. Signal Process.*, 2008, **56**, (4), pp. 1616–1626
12 Gallager, R.G.: 'Low-density parity-check codes' (MIT Press, Cambridge, MA, 1963)
13 Alamri, O., Ng, S.X., Guo, F., Zummo, S., Hanzo, L.: 'Non-binary LDPC-coded sphere-packed transmit diversity', *IEEE Trans. Veh. Technol.*, 2008, **57**, (5), pp. 3200–3205
14 Wymeersch, H., Steendam, H., Moeneclaey, M.: 'Log-domain decoding of LDPC codes over GF(q)'. IEEE Int. Conf. on Communications, June 2004, Vol 2, pp. 772–776
15 Sanchez, O., Jego, C., Jezequel, M.: 'Analysis of the convergence process by EXIT charts for parallel implementations of turbo decoders', *IEEE Commun. Lett.*, 2013, **17**, (7), pp. 1427–1430
16 Ruozzi, N., Tatikonda, S.: 'Message-passing algorithms: re-parameterizations and splitting', *IEEE Trans. Inf. Theory*, 2013, **59**, (9), pp. 5860–5881
17 Burshtein, D., Miller, G.: 'Expander graph arguments for message-passing algorithms', *IEEE Trans. Inf. Theory*, 2001, **47**, (2), pp. 782–790
18 Bruck, J., Ching-Tien, H.: 'Fault-tolerant cube graphs and coding theory', *IEEE Trans. Inf. Theory*, 1996, **42**, (6), pp. 2217–2221
19 Babich, F., Crismani, A., Maunder, R.G.: 'EXIT chart aided design of periodically punctured turbo codes', *IET Electron. Lett.*, 2010, **46**, (14), pp. 1001–1003
20 Kapur, A., Varanasi, M.K., Mullis, C.T.: 'On the limitation of generalized Welch-bound equality signals', *IEEE Trans. Inf. Theory*, 2005, **51**, (6), pp. 2220–2224
21 Nikopour, H., Baligh, H.: 'Sparse code multiple access'. IEEE Int. Symp. on Personal Indoor and Mobile Radio Communications, 2013, pp. 332–336

9 Appendix

9.1 Appendix 1: Flooding schedule of joint detection and decoding

9.1.1 Initialisation: Assuming there is no a priori probability available, initial LLR are set to zeros

$$L_{v_{k,m}\to c_n} = 0, \quad L_{v_{k,m}\to v'_{k,m}} = 0, \quad \forall k, \forall m, \forall n \tag{14}$$

9.1.2 Updating of chip nodes and parity check nodes: LLR of the chip nodes and the parity check nodes are calculated at the

same time. For the chip nodes

$$L_{c_n \to v_{k,m}} = f(v_{k,m}|y_n, L_{v_{k',m'} \to c_n}, (k', m') \in \psi_n \backslash (k, m)) \tag{15}$$

where $\psi_n\backslash(k, m)$ is the set of data symbols (excluding $v_{k,m}$) that interfere on the chip c_n.

In order to approximate the maximum a posteriori probability (MAP) detector, the right-hand side of (15) represents marginalisation function, which is based on (3), and can be written as (see (16))

where $\boldsymbol{v}$ is the transmitted vector, the conditional probability density function $p(y_n/\boldsymbol{v})$ and a priori probability $p_n(v_{k',m'})$ are given as

$$p(y_n|\boldsymbol{v}) = \exp\left(-\frac{1}{2\sigma_A^2}||y_n - \boldsymbol{r}_{[n]}^{\mathrm{T}}\boldsymbol{v}_{[n]}||^2\right) \tag{17}$$

$$p_n(v_{k',m'}) = \exp\left(L_{v_{k',m'} \to c_n}\right) \tag{18}$$

where $\boldsymbol{v}_{[n]}$ and $\boldsymbol{r}_{[n]}$ denote the vectors containing the symbols transmitted by every user that spread its data on the nth chip and their corresponding effective received signature values, respectively. As can be seen from (16), based on the received chip y_n and a priori input information$p_n(v_{k',m'})$, extrinsic values are calculated for all the constituent bits involved in (3). Substituting (17) and (18) into (16), the message update becomes

$$L_{c_n \to v_{k,m}} = \kappa_{n,k,m} \max_{\boldsymbol{v}_{[n]}}^{*}\left(\sum_{(k',m') \in \psi_n \backslash (k,m)} L_{v_{k',m'} \to c_n} - \frac{1}{2\sigma_A^2}||y_n - \boldsymbol{r}_{[n]}^{\mathrm{T}}\boldsymbol{v}_{[n]}||^2\right) \tag{19}$$

where $\kappa_{n,k,m}$ denotes the normalisation coefficient and

$$\max^{*}(a, b) \triangleq \log(\mathrm{e}^a + \mathrm{e}^b) = \max(a, b) + \ln\left(1 + \mathrm{e}^{-|a-b|}\right) \tag{20}$$

In terms of the updating of parity check nodes, we denote the LLR of variable node $v_{k,m}$ as

$$L(v_{k,m}) = \ln\frac{\Pr(v_{k,m} = \sigma)}{\Pr(v_{k,m} = 0)} \tag{21}$$

with $\Pr(v_{k,m} = \sigma)$ representing the probability that $v_{k,m}$ takes on the value σ ($\sigma \in \mathrm{GF}(q)$). For example, suppose we are given the LLR of $v_{1,1}$ and $v_{1,2}$ (abbreviated by L_1 and L_2, respectively) and two elements in GF(q): a_1 and a_2. We calculate $L(a_1v_{1,1} + a_2v_{1,2})$ as (see (22))

where

$$\frac{\Pr(v_{1,1} = a_1^{-1}\sigma)}{\Pr(v_{1,1} = 0)} = \mathrm{e}^{\ln((\Pr(v_{1,1}=a_1^{-1}\sigma))/(\Pr(v_{1,1}=0)))} = \mathrm{e}^{L_1(a_1^{-1}\sigma)} \tag{23}$$

$$\frac{\Pr(v_{1,2} = a_2^{-1}\sigma)}{\Pr(v_{1,2} = 0)} = \mathrm{e}^{\ln(\Pr(v_{1,2}=a_2^{-1}\sigma))/(\Pr(v_{1,1}=0)))} = \mathrm{e}^{L_2(a_2^{-1}\sigma)} \tag{24}$$

Hence, (see (25))

Using (23) and (24), the denominator of (22) can be expressed as (see (26))

Therefore, (see (27))

According to (20)

$$\max^{*}(a, b, c) = \max^{*}(\max^{*}(a, b), c) \tag{28}$$

The output of $L(a_1v_{1,1} + a_2v_{1,2})$ can be recursively calculated by substituting (28) into (27). In order to compute the output LLR of variable nodes, we arrange a permutation by multiplying the

$$f(v_{k,m}|y_n, L_{v_{k',m'} \to c_n}, (k', m') \in \psi_n \backslash (k, m)) = \log\left(\sum p(y_n|\boldsymbol{v})p_n(\boldsymbol{v}|v_{k,m})\right) = \log\left(\sum p(y_n|\boldsymbol{v}) \prod_{(k',m') \in \psi_n \backslash (k,m)} p_n(v_{k',m'})\right) \tag{16}$$

$$\begin{aligned} L(a_1v_{1,1} + a_2v_{1,2}) &= \ln\frac{\Pr(a_1v_{1,1} + a_2v_{1,2} = \sigma)}{\Pr(a_1v_{1,1} + a_2v_{1,2} = 0)} = \ln\frac{\sum_{x \in \mathrm{GF}(q)} \Pr(v_{1,1} = x)\Pr(v_{1,2} = a_2^{-1}(\sigma + a_1x))}{\sum_{y \in \mathrm{GF}(q)} \Pr(v_{1,1} = y)\Pr(v_{1,2} = a_2^{-1}a_1y)} \\ &= \ln\frac{\sum_{x \in \mathrm{GF}(q)} \dfrac{\Pr(v_{1,1} = x)\Pr(v_{1,2} = a_2^{-1}(\sigma + a_1x))}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)}}{1 + \sum_{y \in \mathrm{GF}(q)/0} \dfrac{\Pr(v_{1,1} = y)\Pr(v_{1,2} = a_2^{-1}a_1y)}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)}} \\ &= \ln\frac{\dfrac{\Pr(v_{1,1} = a_1^{-1}\sigma)}{\Pr(v_{1,1} = 0)} + \dfrac{\Pr(v_{1,2} = a_2^{-1}\sigma)}{\Pr(v_{1,2} = 0)} + \sum_{x \in \mathrm{GF}(q)/\{0, a_1^{-1}\sigma\}} \dfrac{\Pr(v_{1,1} = x)\Pr(v_{1,2} = a_2^{-1}(\sigma + a_1x))}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)}}{1 + \sum_{y \in \mathrm{GF}(q)/0} \dfrac{\Pr(v_{1,1} = y)\Pr(v_{1,2} = a_2^{-1}a_1y)}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)}} \end{aligned} \tag{22}$$

$$\sum_{x \in \mathrm{GF}(q)/\{0, a_1^{-1}\sigma\}} \frac{\Pr(v_{1,1} = x)\Pr(v_{1,2} = a_2^{-1}(\sigma + a_1x))}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)} = \sum_{x \in \mathrm{GF}(q)/\{0, a_1^{-1}\sigma\}} \mathrm{e}^{L_1(x) + L_2(a_2^{-1}(\sigma + a_1x))} \tag{25}$$

$$1 + \sum_{y \in GF(q)/0} \frac{\Pr(v_{1,1} = y)\Pr(v_{1,2} = a_2^{-1}a_1x)}{\Pr(v_{1,1} = 0)\Pr(v_{1,2} = 0)} = 1 + \sum_{y \in GF(q)/0} \mathrm{e}^{L_1(y) + L_2(a_2^{-1}a_1y)} \tag{26}$$

$$L(a_1v_{1,1} + a_2v_{1,2}) = \ln\left(\mathrm{e}^{L_1(a_1^{-1}\sigma)} + \mathrm{e}^{L_2(a_2^{-1}\sigma)} + \sum_{x \in \mathrm{GF}(q)/\{0, a_1^{-1}\sigma\}} \mathrm{e}^{L_1(x) + L_2(a_2^{-1}(\sigma + a_1x))}\right) - \ln\left(1 + \sum_{y \in \mathrm{GF}(q)/0} \mathrm{e}^{L_1(y) + L_2(a_2^{-1}a_1y)}\right) \tag{27}$$

non-zero values, that is

$$L_{v'_{k,m}\to p_{k,j}} = L_{v_{k,m}h_{k,m}\to p_{k,j}} \tag{29}$$

Furthermore, partial sums are declared as

$$\alpha_{ki} = \sum_{m:m\le i} h_{k,m}v_{k,m} \tag{30}$$

$$\beta_{ki} = \sum_{m:m\ge i} h_{k,m}v_{k,m} \tag{31}$$

The LLR of these partial sums are

$$L(\alpha_{ki}) = L\left(\sum_{m:m\le i-1} h_{k,m}v_{k,m} + h_{k,i}v_{k,i}\right) \tag{32}$$

$$L(\beta_{ki}) = L\left(\sum_{m:m\ge i+1} h_{k,m}v_{k,m} + h_{k,i}v_{k,i}\right) \tag{33}$$

Therefore the LLR of the parity check node can be expressed as

$$\begin{aligned} L(\alpha_{ki}) &= L\left(\sum_{m:m\le i-1} h_{k,m}v_{k,m} + h_{k,i}v_{k,i}\right) \\ &= \ln\frac{\Pr(h_{k,m}^{-1}\alpha_{k(m-1)} + h_{k,m}^{-1}\beta_{k(m+1)} = t)}{\Pr(h_{k,m}^{-1}\alpha_{k(m-1)} + h_{k,m}^{-1}\beta_{k(m+1)} = 0)} \\ &= L(h_{k,m}^{-1}\alpha_{k(m-1)} + h_{k,m}^{-1}\beta_{k(m+1)}) \end{aligned} \tag{34}$$

9.1.3 Updating of variable nodes and permutation nodes:

In a single graph, variable nodes only gather information from one type of node (chip node or parity check node). However, in the JSG, the updating of $L_{v_{k,m}\to c_n}$ not only receives chip node information, but also utilises the information that comes from permutation nodes in the right dotted box in Fig. 1*b*, that is

$$L_{v_{k,m}\to c_n} = \sum_{n'\in\varepsilon_{k,m}\backslash n} L_{c_{n'}\to v_{k,m}} + \sum_{j\in\omega_{k,m}} L_{p_{k,j}\to v_{k,m}} \tag{35}$$

with $\varepsilon_{k,m}\backslash n$ representing the set of chips (excluding c_n) that $v_{k,m}$ is spread on, and

$$L_{p_{k,j}\to v_{k,m}} = L_{p_{k,j}\to v'_{k,m}/h_{k,m}} \tag{36}$$

where $v'_{k,m}/h_{k,m}$ representing $v'_{k,m}$ divided by $h_{k,m}$ over GF(q).

Similarly, updating of $L_{v_{k,m}\to p_{k,j}}$ also involves the information from both sides, that is

$$L_{v_{k,m}\to p_{k,j}} = \sum_{j'\in\omega_{k,m}\backslash j} L_{p_{k,j'}\to v'_{k,m}} + \sum_{n\in\omega_{k,m}} L_{c_n\to v_{k,m}} \tag{37}$$

with $\omega_{k,m}\backslash j$ representing the set of parity check nodes (excluding $p_{k,j}$) that connect to the variable node $v_{k,m}$.

9.1.4 Estimation and syndrome computing:

In the single graph case of LDS-CDMA, a posterior probability of the transmitted symbol can only be calculated after a fixed number of iterations, as there is no criterion to determine whether the iterative message has converged. Fortunately, in the joint low density graph, parity check nodes are available, thus it is possible to terminate the joint detection and decoding by syndrome computing. A posterior probability of the transmitted symbol $v_{k,m}$ is calculated as

$$L_{v_{k,m}} = \sum_{n\in\varepsilon_{k,m}} L_{c_n\to v_{k,m}} + \sum_{j\in\omega_{k,m}} L_{p_{k,j}\to v_{k,m}} \tag{38}$$

The estimated value of the variable node $v_{k,\,m}$ is obtained by making a hard decision

$$\hat{v}_{k,m} = \arg\max_{v_{k,m}} L_{v_{k,m}} \tag{39}$$

If the result of syndrome computing equals to zero or the maximum iteration number is reached, the process is terminated. Otherwise, the iteration goes on.

9.2 Appendix 2: Serial schedule of joint detection and decoding

9.2.1 Initialisation:

$$L^1_{v_{k,m}\to c_n} = 0, \quad L^1_{v_{k,m}\to v'_{k,m}} = 0, \ \forall k, \ \forall m, \ \forall n \tag{40}$$

9.2.2 Updating of chip nodes sequentially:

1. Accumulating all the messages delivered to the chip node c_n

$$D = \sum_{(k,m)\in\psi_n} \left(L^{i-1}_{v_{k,m}} - L^{i-1}_{c_n\to v_{k,m}}\right) \tag{41}$$

2. For each variable node that is connected to the chip node c_n

$$E = L^{i-1}_{v_{k,m}} - L^{i-1}_{c_n\to v_{k,m}} \tag{42}$$

$$L^i_{c_n\to v_{k,m}} = \kappa_{n,k,m} \max^{*}_{\mathbf{v}_{[n]}}\left(D - E - \frac{1}{2\sigma_A^2}||y_n - \mathbf{r}^{\mathrm{T}}_{[n]}\mathbf{v}_{[n]}||^2\right) \tag{43}$$

$$L^i_{p_{k,j}\to v_{k,m}} = L\left(h_{k,m}^{-1}\alpha_{k(m-1)} + h_{k,m}^{-1}\beta_{k(m+1)}\right) \tag{36}$$

$$L^i_{v_{k,m}} = E + L^i_{c_n\to v_{k,m}} + L^i_{p_{k,j}\to v_{k,m}} \tag{44}$$

9.2.3 Estimation and syndrome computing:

The estimated value of the variable node $v_{k,m}$ is estimated by (29), and syndrome computation is done to determine whether iteration is stopped.

IET Commun., 2015, Vol. 9, Iss. 5, pp. 707–718