www.ietdl.org

Published in IET Communications
Received on 29th August 2008
Revised on 30th January 2009
doi: 10.1049/iet-com.2008.0503

ISSN 1751-8628

Application of complex-network theories to the design of short-length low-density-parity-check codes

X. Zheng[1] F.C.M. Lau[1] C.K. Tse[1] Y. He[2] S. Hau[1]

[1]Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong
[2]Department of Communication Engineering, College of Information Engineering, Shenzhen University, People's Republic of China
E-mail: encmlau@polyu.edu.hk

Abstract: Study of complex networks has been conducted across many fields of science, including computer networks, biological networks and social networks. Characteristics of different types of complex networks such as random networks, regular-coupled networks, small-world networks and scale-free networks have been discovered by researchers. Application of such network properties to solve engineering problems, however, is still at the infancy stage. In this study, we make one of the first attempts in applying complex network theories to communications engineering. In particular, inspired by the shortest-average-path-length property of scale-free networks, we design short-length low-density-parity-check (LDPC) codes with an aim to shortening the average distance between any two variable nodes. We will also compare the error performance, both theoretically and by simulations, of the proposed codes with those of other well-known LDPC codes.

1 Introduction

In recent years, complex networks have been studied across many fields of science, including computer networks, biological networks, social networks, power networks and telephone call networks [1–6]. Synchronisation and stability of different complex dynamical networks have also been gaining much research interest in the engineering discipline [7–11].

A typical complex network is composed of nodes together with the connections (links) between them [12, 13]. For example, in an acquaintance network, each individual person is represented by a node and a connection is established between two nodes if the corresponding individuals are acquaintances (such as relatives, friends and colleagues) of each other. Suppose the network contains N people. We can denote the topology of the acquaintance network by a connection matrix $\boldsymbol{R}$ given by

$$\boldsymbol{R} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,N} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N,1} & R_{N,2} & \cdots & R_{N,N} \end{bmatrix} \tag{1}$$

where

$$R_{ij} = \begin{cases} 1 & \text{if Individual } i \text{ is an acquaintance of Individual } j \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Moreover, the connection matrix has the following properties.

1. $R_{ii} = 0$ because Individual i is not regarded as an acquaintance of himself.

2. $R_{ij} = R_{ji}$ because either the ith and jth individuals are or are not acquaintances of each other.

3. The sum of elements in the ith row or column gives the total number of acquaintances for the ith individual (denoted by n_i), i.e.

$$n_i = \sum_{j=1}^{N} R_{ij} = \sum_{j=1}^{N} R_{ji} \quad (3)$$

It is also called the *degree* of the ith node.

4. The average sum of elements in each row (or column) gives the average number of acquaintances of an individual in the network (denoted by $\overline{n}$), i.e.

$$\overline{n} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} R_{ij} \quad (4)$$

5. If the total number of acquaintances for the ith individual equals $\overline{n}$ for $i = 1, 2, \ldots, N$, the network is called a *uniform complex network*. Otherwise, the complex network is *non-uniform*.

Furthermore, the distance between two nodes in a network is defined as the number of edges along the shortest path connecting them. The average path length (APL) of the whole network is then the mean distance between any two nodes, i.e. averaging the distance over all pairs of nodes.

There are four well-known and much studied classes of complex networks: random network, regular lattice, small-world network and scale-free network [12, 13]. Consider a network consisting of N nodes. A random network is constructed when an edge is added to each pair of nodes with a certain probability. In a random network, the APL between nodes is proportional to $\log(N)$ while the degree of the nodes follows a Poisson distribution. To construct a regular lattice, the N nodes are first spaced equally on a circle and each of the nodes is then connected to a given number of adjacent nodes. The APL of a regular lattice is relatively large and becomes infinite as N approaches infinity. However, most of the real-world networks are neither entirely regular nor entirely random. Suppose we start with a regular lattice with N nodes and with a probability p, we rewire each of the links randomly. Due to the rewiring process, the APL of the network becomes shorter [proportional to $\log(N)$] compared with that of the lattice. Hence, we end up with a 'small-world' network because the average distance between the nodes are 'smaller' [1]. The probability distribution of the node degrees of a small-world network is determined by the rewiring probability p. In the extreme cases of $p = 0$ and $p = 1$, a regular lattice and a random network are formed, respectively.

A recent significant discovery in the complex network theory is that some complex networks, such as the Internet and the worldwide web, have their node degrees following power-law distributions, i.e.

$$\Pr(n_i) = A n_i^{-\gamma} \quad (5)$$

where $\Pr(n_i)$ denotes the probability of a randomly selected node having a degree n_i, γ is the characteristic exponent and A is the normalising coefficient. Such kind of networks are called *scale-free networks* [2]. Fig. 1 plots the probability distribution of n_i of a scale-free network. From this graph, we can observe that the majority of the nodes have a few connections, whereas a small number of nodes (sometimes referred to as *super nodes*) have a large number of connections. Fig. 2 illustrates another scale-free network with 100 nodes. Note that the five super nodes (represented by nodes 1–5) make connections to most of the nodes. Compared with regular-coupled networks, small-world networks and random networks, scale-free networks of the same size (number of nodes) and with the same number of connections are found to accomplish the shortest APL [14]. Whereas for the same APL, among the aforementioned networks, complex networks with scale-free property have the smallest number of connections. It has also been shown that when the value of the characteristic exponent, i.e. γ, lies between 2 and 3, the APL of scale-free networks is $O(\log(\log(N)))$ [14].

In this paper, we exploit the shortest-APL property of scale-free networks and apply it to the design of short-length low-density-parity-check LDPC codes. Specifically, we will propose constructing short-length LDPC codes with variable-node degrees following power-law distributions. Here, we refer such LDPC codes to as *scale-free LDPC* (SF-LDPC) codes. We will compare the achievable error

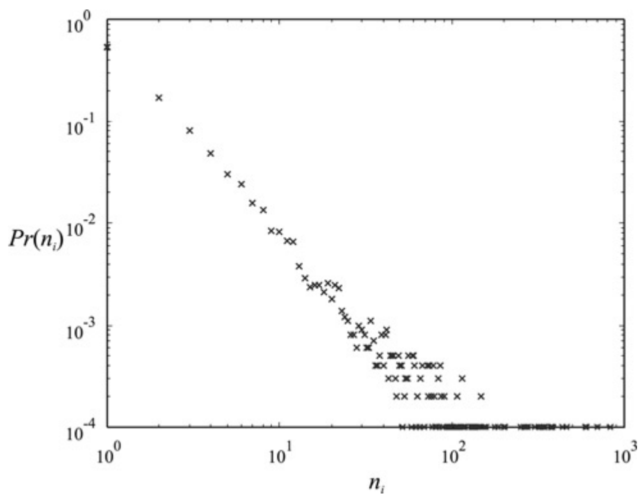

Figure 1 *Power-law probability distribution. $\bar{n} = 5$ and $\gamma = 2.1$.*

doi: 10.1049/iet-com.2008.0503

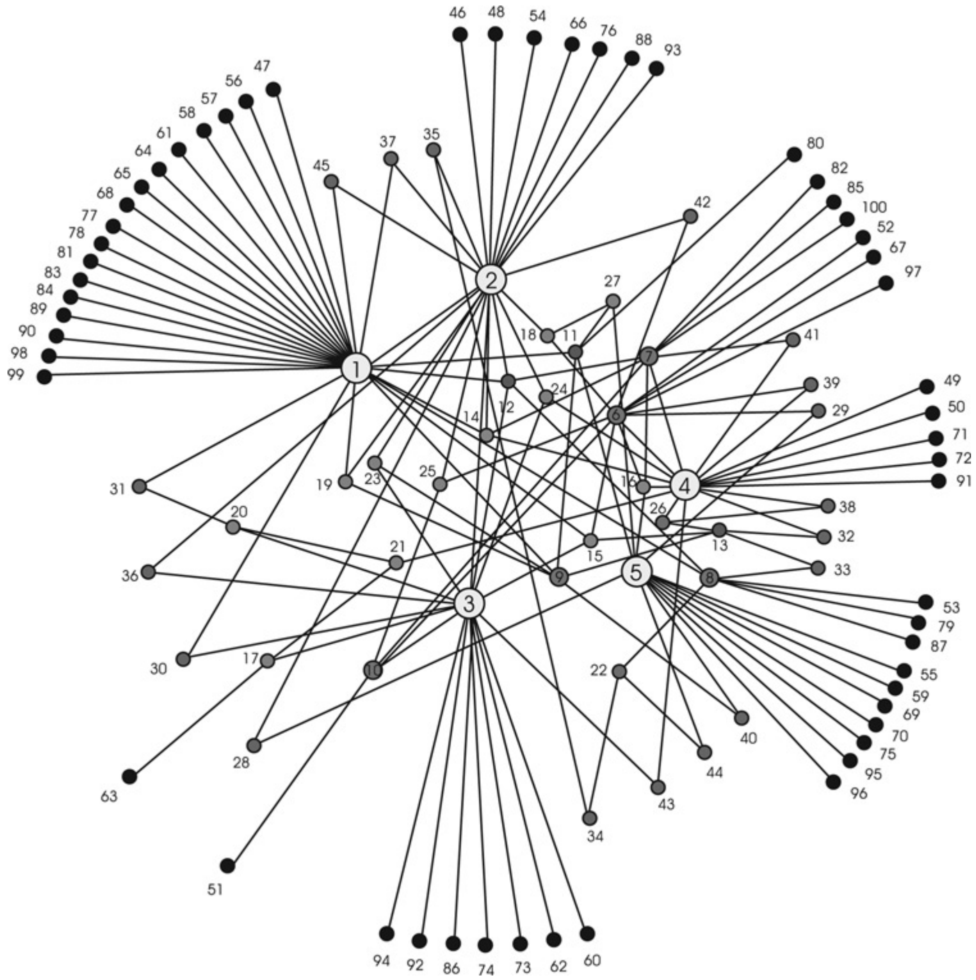

Figure 2 *Scale-free user network with 100 nodes. $\bar{n}$ = 2.9*

performance (threshold) and the complexity (in terms of average number of node degrees) between the proposed short-length SF-LDPC codes and other best-known LDPC codes. Moreover, we will construct SF-LDPC codes of length 2016, 1008 and 504 and simulate their error performance under an additive white Gaussian noise (AWGN) channel environment. Finally, we will compare the error rates and the average convergence time between the constructed SF-LDPC codes and some other best-known LDPC codes.

2 Density evolution and LDPC codes

In the bipartite graph representation of LDPC codes, the code bits and the parity-check equations are denoted by two kinds of nodes, namely variable nodes and check nodes, respectively. Also, the variable nodes and check nodes are connected by edges, which are governed by the entries in the sparse parity-check matrix. For each node, the number of edges connected is called the 'degree' of the node. If all nodes of the same type have the same degree, the LDPC codes are regular; otherwise, the codes are irregular. For a given distribution pair (λ, ρ) of an LDPC ensemble [15],

$$\lambda(x) := \sum_{k=2}^{d_v} \lambda_k x^{k-1} \tag{6}$$

and

$$\rho(x) := \sum_{k=2}^{d_c} \rho_k x^{k-1} \tag{7}$$

specify, respectively, the variable-node and check-node degree distributions. Also, d_v is the maximum variable-node degree and d_c denotes the maximum check-node degree. Moreover, the coefficients λ_k and ρ_k, respectively, represent the fraction of edges connected to the variable and check nodes with degree k.

Assume that the parity-check matrix has a full rank. Based on the degree distributions, the code rate of the system, denoted by R_{code}, can be obtained using

$$R_{\text{code}} = 1 - \int_0^1 \rho(x)\,dx \Big/ \int_0^1 \lambda(x)\,dx. \tag{8}$$

Fig. 3*a* shows an example of a (10, 5) irregular LDPC code where 10 and 5 denote the number of variable nodes and check nodes, respectively. In this case, the code rate can also be derived using $R_{\text{code}} = (10 - 5)/10 = 0.5$. In other words, for a finite-length LDPC code, the code rate can be determined directly from

$$R_{\text{code}} = 1 - L/N \tag{9}$$

where L and N represent, respectively, the number of check nodes and the number of variable nodes.

In the design of LDPC codes, the error performance of the codes has been one of the major considerations. Suppose an LDPC code is defined with its degree distributions of the variable nodes and the check nodes. Moreover, the most common algorithm, namely the sum-product iterative decoding algorithm or the belief propagation (BP) algorithm [16], is used in the decoder. In the decoding process, updated messages will be passed forward and backward between the variable nodes and the check nodes during each iteration. Under an AWGN channel with noise power σ^2, the *best achievable performance* of the code will be determined by a certain 'threshold', which can be regarded as the maximum noise standard deviation σ^* below which error-free communication can always be achieved. Hence, the larger the threshold value, the better theoretical performance the code achieves.

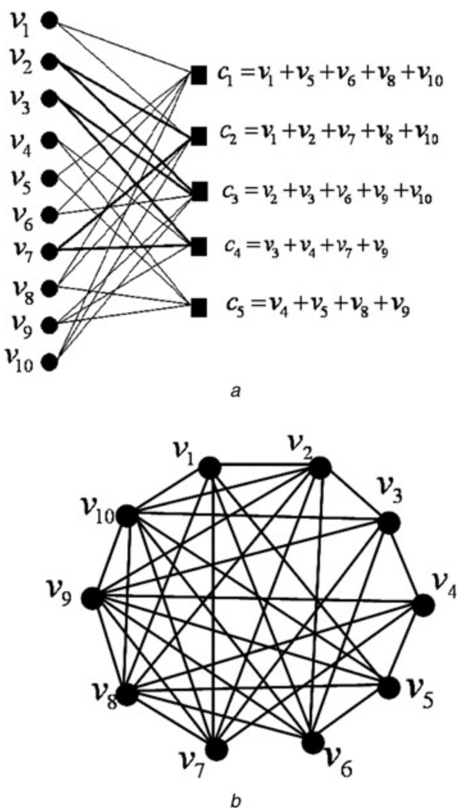

Figure 3 *(10, 5) LDPC code*

a Bipartite graph representation
b Complex network formed by removing the check nodes

To evaluate the threshold value, a powerful tool called 'density evolution (DE)' has been proposed [16, 17]. Though a complex algorithm, DE can compute the exact threshold of a code with arbitrarily small error probability. In contrast, other popular evaluation tools, including 'Gaussian approximate density evolution' (GA-DE) [18], 'extrinsic information transfer' (EXIT) [19] and 'generalised extrinsic information transfer' (GEXIT) [20], provide simpler methods to compute the threshold at the expense of accuracy.

With the use of the aforementioned evaluation tools, researchers have been able to optimise the achievable error performance of the LDPC codes under different channel conditions by varying the parameters of the codes, including variable-node distribution, check-node degree distribution, maximum variable-node degree, maximum check-node degree and code rate (a function of node degree distributions) [16, 17, 19]. However, the best achievable error performance can only be accomplished under two conditions – infinite code length and infinite number of iterations performed by the decoder – and neither requirement can be fulfiled in practice. In reality, short-length (less than several thousands) LDPC codes will find a lot more applications, but their error performance may deviate significantly from the best values. So, other degree distributions, compared with those optimised by the aforementioned tools, may possibly offer a better performance.

3 SF-LDPC codes

Recall that as the iterative decoding process proceeds, the information generated by each variable node will eventually be conveyed to all other variable nodes via the check nodes. To visualise the flow of messages among the variable nodes, we remove the check nodes and construct a complex network using only the variable nodes. Moreover, two variable nodes v_i and v_j are connected in the complex network only if they are connected to the same check node in the original bipartite graph. Fig. 3*b* shows the complex network formed based on the LDPC code in Fig. 3*a*.

For the complex network so formed, the APL corresponds to the average number of iterations required for an updated message from one variable node to eventually pass to another variable node. Decreasing the APL will no doubt accelerate the exchange of messages among the variable nodes, thereby reducing the number of iterations the decoding algorithm takes to converge. This is particularly useful in the high SNR region where the decoder has a much higher chance to converge. As scale-free networks have been shown to provide a very short APL [14], it will be an advantage if LDPC codes are designed in such a way that the resultant complex network formed by the variable nodes has a scale-free property.

doi: 10.1049/iet-com.2008.0503

One approach is to start with a complex network (like Fig. 3*b*) with a power-law degree distribution and convert it directly into a bipartite graph (like Fig. 3*a*) that represents the LDPC code. However, the conversion task is not a trivial one as it can be envisaged that the mapping from the complex network to the bipartite graph is not unique. An alternate way is to begin with a bipartite graph and convert it into a complex network. The challenge would then become ensuring that the complex network has a scale-free property. It is because the variable nodes are interconnected via the check nodes and hence it may not be possible to determine the properties of the resultant complex network when the check nodes are removed. To resolve the issue, we apply a theorem in [21], which states that if the degree distribution of one set of nodes in a bipartite graph follows a power-law distribution, the degree distribution of the unipartite graph (network) formed when the other set of nodes is removed also follows a power-law with the same exponent. In other words, if we can construct LDPC codes such that their variable-node degrees follow power-law distributions, the complex networks formed by the variable nodes alone (after removing the check nodes) will also follow power-law distributions with the same exponent. Consequently, the APL between the variable nodes will be small which would enhance the convergence rate of the LDPC decoder.

We denote the probability that a variable node has k connections by $\Pr_\lambda(k)$. To construct a SF-LDPC code, we assign the fraction of variable nodes with degree k according to a power-law function, i.e. $\Pr_\lambda(k) \propto k^{-\gamma}$, where γ is the characteristic exponent for the variable-node degree. Since

$$\sum_k \Pr_\lambda(k) = 1 \tag{10}$$

the fraction of edges connecting to variable nodes with degree k can be readily shown equal to

$$\lambda_k = \frac{k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{1-\gamma}} \tag{11}$$

Then the variable-node degree distribution in (9) can be expressed as

$$\lambda(x) = \sum_{k=2}^{d_v} \frac{k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{1-\gamma}} x^{k-1} \tag{12}$$

In addition, the average variable-node degree, denoted by $\langle k_v \rangle$, can be computed from

$$\langle k_v \rangle = \frac{\sum_{k=2}^{d_v} k^{1-\gamma}}{\sum_{i=2}^{d_v} i^{-\gamma}} \tag{13}$$

As for the check nodes, their degree distributions are also found to affect the performance of the LDPC codes to some extent. Moreover, it has been well known that the degrees of the check nodes should be kept almost the same in the design of good LDPC codes [16]. Here, we restrict the check-node degrees to three consecutive integers, i.e. $d_c - 2$, $d_c - 1$ and d_c and that the check-node degrees are taken to follow a Poisson distribution with parameter μ. The advantage of such a model is that there are only two variables to manipulate – d_c and μ. Consolidating the above conditions, the probability that a check node has $k \in \{d_c - 2, d_c - 1, d_c\}$ connections, denoted by $\Pr_\rho(k)$, equals

$$\Pr_\rho(k) = \frac{\mu^k e^{-\mu}/k!}{\sum_{k'=d_c-2}^{d_c} \mu^{k'} e^{-\mu}/k'!} \tag{14}$$

Then, the fraction of edges connecting to check nodes with degree k equals

$$\rho_k = \frac{\mu^k e^{-\mu}/(k-1)!}{\sum_{j=d_c-2}^{d_c} \mu^j e^{-\mu}/(j-1)!} \quad k \in \{d_c - 2, d_c - 1, d_c\} \tag{15}$$

and the check-node degree distribution in (10) can be rewritten as

$$\rho(x) = \sum_{k=d_c-2}^{d_c} \frac{\mu^k e^{-\mu}/(k-1)!}{\sum_{j=d_c-2}^{d_c} \mu^j e^{-\mu}/(j-1)!} x^{k-1} \tag{16}$$

Combining the results in (8), (12), (13) and (15), it can be readily shown that for a given rate R_{code}

$$\begin{aligned} \frac{\langle k_v \rangle}{1 - R_{code}} &= \frac{\sum_{k=d_c-2}^{d_c} \mu^k e^{-\mu}/(k-1)!}{\sum_{j=d_c-2}^{d_c} \mu^j e^{-\mu}/j!} \\ &= \frac{(d_c-2)(d_c-1)d_c + (d_c-1)d_c\mu + d_c\mu^2}{(d_c-1)d_c + d_c\mu + \mu^2} \end{aligned} \tag{17}$$

Since d_c is an integer greater than 2, we can conclude that

$$d_c - 2 < \frac{\langle k_v \rangle}{1 - R_{code}} < d_c \tag{18}$$

and

$$d_c = \left\lceil \frac{\langle k_v \rangle}{1 - R_{code}} \right\rceil, \left\lceil \frac{\langle k_v \rangle}{1 - R_{code}} \right\rceil + 1 \tag{19}$$

where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. When d_c is selected, the corresponding μ can also be found using (17). Supposing $d_v = 20$ and $R_{code} = 0.5$, Table 1 shows the possible values of d_c as γ varies.

Table 1 Possible values of d_c at different ranges of γ. $d_v = 20$ and $R_{code} = 0.5$

Range of γ	Possible values of d_c
1.727−1.934	10,11
1.934−2.183	9,10
2.183−2.501	8,9
2.501−2.963	7,8
2.963−3.830	6,7

4 Results and discussions

4.1 Achievable error performance of SF-LDPC codes

In this section, we present the analytical performance and the simulated results for SF-LDPC codes. First, we compare the achievable error-correcting capability (threshold) between SF-LDPC codes and other best-known LDPC codes [22].

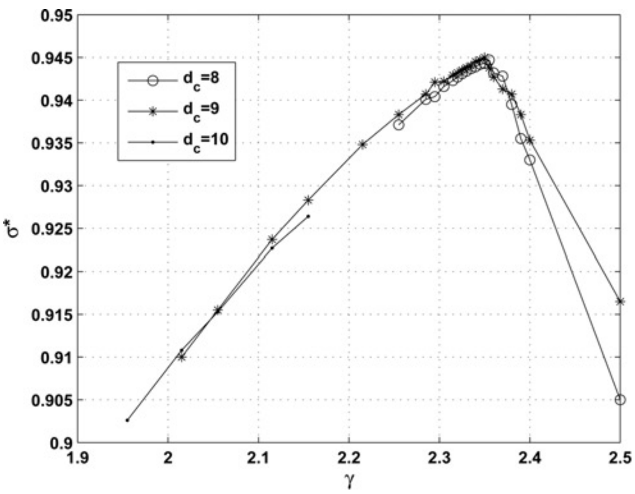

Figure 4 *Achievable error performance (threshold) σ^* against γ. $R_{code} = 0.5$ and $d_v = 20$*

We assume a code rate of 0.5 and an AWGN channel. Suppose the maximum variable-node degree equals 20, i.e. $d_v = 20$. We select a value for γ, say $\gamma = 2.0$. Based on (19) and (17), we can find the values of d_c and μ, respectively. We then substitute the distributions of the variable nodes and check nodes into the DE algorithm, which can be treated as a black box here, and obtain the achievable error performance σ^* of the SF-LDPC codes. Fig. 4 plots the value of σ^* as γ increases from 1.95 to 2.50. From the results, we observe that σ^* accomplishes a maximum value of 0.945 at $\gamma = 2.35$ and $d_c = 9$. Note that $d_c = 8$ and 10 are, respectively, valid only in the ranges $\gamma \in [2.18, 2.50]$ and $\gamma \in [1.95, 2.18]$. But $d_c = 9$ is valid in the range $\gamma \in [1.95, 2.50]$ (see Table 1 for reference).

Using the same methodology, we can evaluate the best achievable error performance (threshold) σ^* of the SF-LDPC codes with code rate $R_{code} = 0.5$ under different values of d_v. In Table 2, we list the highest thresholds achieved by SF-LDPC codes and the corresponding parameters used, alongside with the thresholds of other best-known LDPC codes [22]. It can be observed that in all cases, the largest threshold values σ^* for the SF-LDPC codes are comparable with those for other best-known LDPC codes (less than 2% difference). But the average number of connections for the SF-LDPC codes is significantly smaller (12 to 15% reduction) compared to those for other LDPC codes. In the following studies, all the SF-LDPC codes will therefore be constructed based on the parameters listed in Table 2.

4.2 Characteristics of short-length SF-LDPC codes

Next, we form SF-LDPC codes of finite lengths using the parameters listed in Table 2. We select two codes randomly from the SF-LDPC code ensemble. The first one has a block length of 1000 (i.e. number of variable nodes equals 1000) and a maximum variable-node degree of $d_v = 15$ whereas the other one has a block length of 10 000 and a maximum variable-node degree of $d_v = 20$. Then, we remove the check nodes and form complex networks with the remaining variable nodes, like the one shown in

Table 2 Comparison of threshold value and average number of connections between SF-LDPC codes and other best-known LDPC codes. Code rate $R_{code} = 0.5$

Common parameters	Optimised codes in [22]		SF-LDPC codes				
d_v	σ^*	$\langle k_v \rangle$	γ	d_c	μ	σ^*	$\langle k_v \rangle$
15	0.9622	4.0087	2.35	9	7.3545	0.9430	3.5117
20	0.9649	4.1638	2.35	9	7.8441	0.9449	3.7192
30	0.9690	4.4963	2.36	9	8.5609	0.9513	3.9723
50	0.9718	5.0765	2.35	9	9.4169	0.9535	4.3039

IET Commun., 2009, Vol. 3, Iss. 10, pp. 1569–1577
doi: 10.1049/iet-com.2008.0503

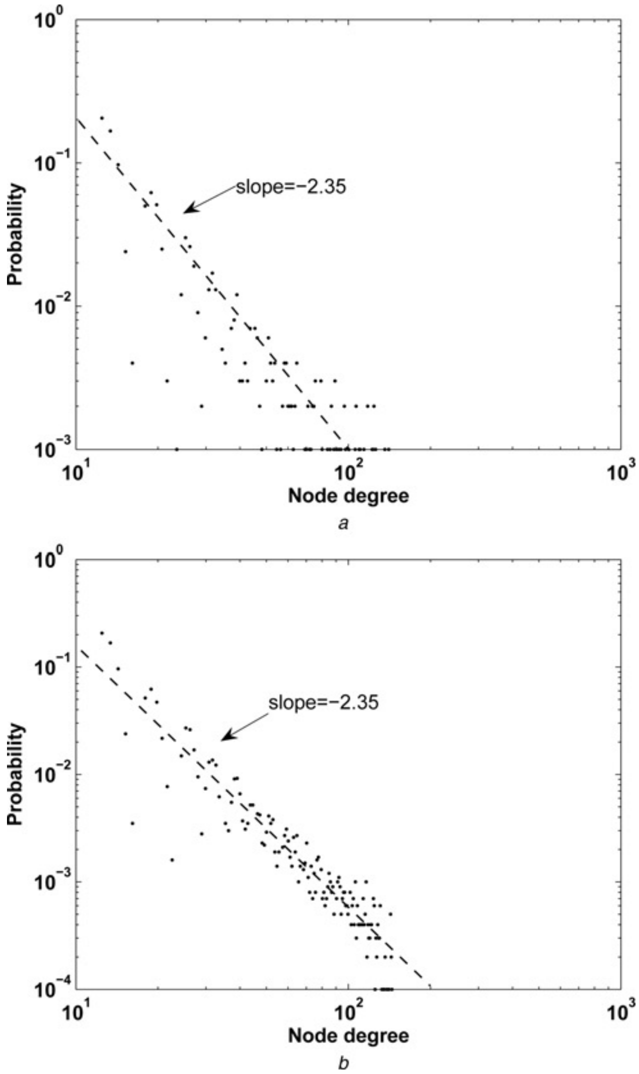

Figure 5 *Degree distributions of the complex variable-node networks of the two randomly selected codes*

a Block length 1000 and maximum variable-node degree 15
b Block length 10000 and maximum variable-node degree 20

Fig. 3*b*. Fig. 5 shows the degree distributions of the complex variable-node networks of the two randomly selected codes. We observe that the degree distributions of the resultant network do follow power-laws with characteristic exponents equalling 2.35. Moreover, when the code length becomes longer, the scale-free property gets more prominent.

Table 3 Details of LDPC code types used in simulations

Abbreviation	Type of code	d_v	σ^*	$\langle k_v \rangle$
DE10	DE-optimised codes in [22]	10	0.9558	3.6631
DE15	DE-optimised codes in [23]	15	0.9622	4.0087
SF20	SF-LDPC codes	20	0.9449	3.7192

4.3 Error performance of short-length SF-LDPC codes

Finally, we compare the simulated error performance of three short-length LDPC codes. Details of the codes are given in Table 3. The first two code types, abbreviated

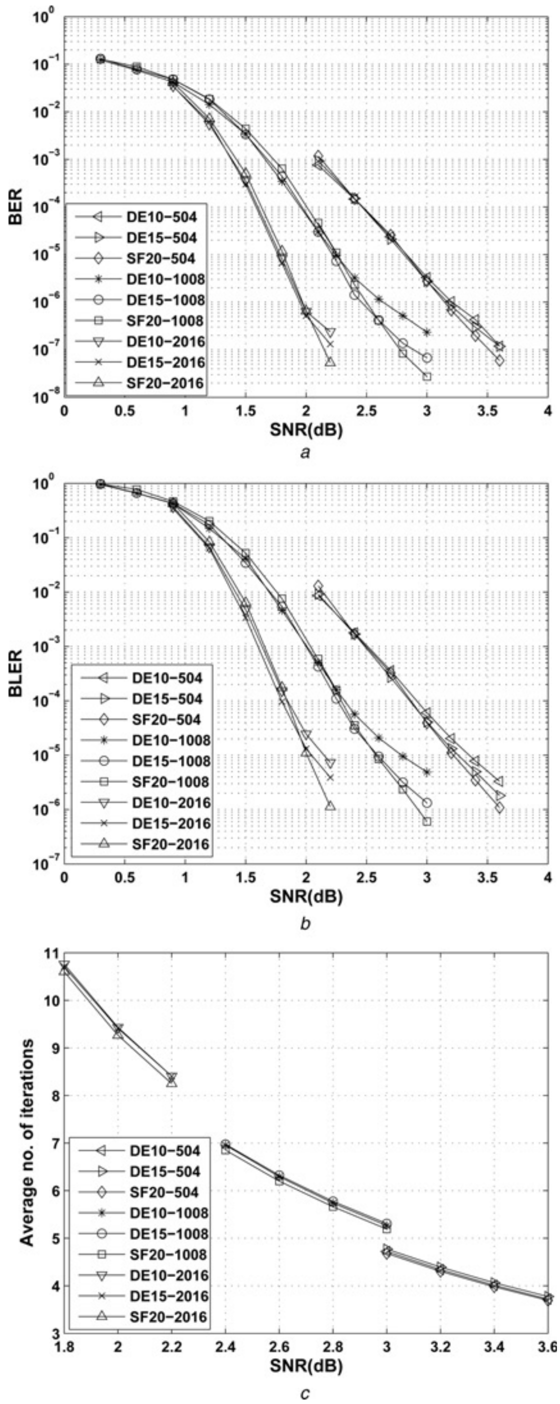

Figure 6 *Performance of three different types of LDPC codes – 'DE10', 'DE15' and 'SF20'. Code lengths are 2016, 1008 and 504 whereas the code rate is 0.5*

a Bit error rate
b Block error rate
c Average number of iterations to decode a codeword (m)

Table 4 Comparison of average convergence times

SNR/code length	Code type	m	$\langle k_v \rangle$	$t_c = m \times \langle k_v \rangle$	Normalisd t_c
3.6 dB/504	DE10/DE15/SF20	3.72/3.78/3.69	3.65/4.00/3.70	13.59/15.09/13.68	0.99/1.10/1.00
3.0 dB/1008	DE10/DE15/SF20	5.27/5.31/5.19	3.66/4.00/3.71	19.26/21.23/19.26	1.00/1.10/1.00
2.2 dB/2016	DE10/DE15/SF20	8.41/8.41/8.25	3.66/4.01/3.71	30.78/33.67/30.64	1.00/1.10/1.00

by 'DE10' and 'DE15', are LDPC codes of which the degree distributions are purely optimised by the DE algorithm [22, 23]. The third code, abbreviated by 'SF20', is our proposed SF-LDPC code. The variable nodes and the check nodes for 'DE10' and 'SF20' codes are connected using the progressive-edge-graph (PEG) method [24], which has been shown to produce codes with both large girth and large Hamming distance. For the codes denoted by 'DE15', we directly apply the codes constructed in [23], which are the best-known LDPC codes in terms of error performance that possess the properties listed in Table 3.

Three different code lengths are used – 2016, 1008 and 504, while the code rate is kept at 0.5. The maximum number of iterations performed to decode one codeword is limited to 50 and the decoding process will be terminated once the maximum number is reached. In Figs. 6*a* and 6*b*, we plot the simulated bit error rates (BERs) and block error rates (BLERs), respectively, for the three types of codes under study. It can be observed that the SF-LDPC codes 'SF20' provide similar BER and BLER performance as 'DE10' and 'DE15' codes at low SNR, and outperform them at higher SNR values. In addition, Fig. 6*c* depicts that SF-LDPC codes 'SF20' can be decoded with a slightly smaller number of iterations, on average, compared with the other DE-optimised codes.

To further compare the performance of the codes, we define the metric 'average convergence time', denoted by t_c, as the product of the average number of iterations to converge (m) and the average variable-node degree ($\langle k_v \rangle$). In general, the smaller the 'average convergence time', the less time the decoder takes to decode a codeword. Table 4 shows the typical results for the 'DE10', 'DE15' and 'SF20' codes for the code lengths 504, 1008 and 2016. It indicates that 'DE10' and 'SF20' have almost identical 'average convergence time' whereas 'DE15' requires, on average, 10% more time (resources) to decode a codeword. But recall that 'SF20' produces less errors than 'DE10' and 'DE15' at higher SNR values

5 Conclusion

In this paper, we have made one of the first attempts in applying complex network theories to solving engineering problems. In particular, we have proposed a new concept in designing short-length LDPC codes with very good performance. We have shown that theoretically, our proposed SF-LDPC codes can accomplish very similar achievable error performance (threshold) compared with DE-optimised LDPC codes. Moreover, we have constructed some short-length SF-LDPC codes and simulate their error performance under an AWGN channel. The results have shown that SF-LDPC codes outperform other DE-optimised codes, producing lower block/bit error rates at high SNRs. Finally, we defined a metric to compare the average convergence times of the various LDPC codes. We further conclude that the convergence time of the proposed SF-LDPC codes is no worse than those of the DE-optimised codes.

6 Acknowledgment

This work was supported by a research grant (G-YG34) provided by Hong Kong Polytechnic University.

7 References

[1] WATTS D.J., STROGATZ S.H.: 'Collective dynamics of 'small-world' networks', *Nature*, 1998, **393**, pp. 440–442

[2] BARABÁSI A.-L., ALBERT R.: 'Emergence of scaling in random networks', *Science*, 1999, **286**, (5439), pp. 509–512

[3] CHEN L., LU J., LU J.: 'Synchronization of the time-varying discrete biological networks'. Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2007), New Orleans, USA, May 2007, pp. 2650–2653

[4] GOU B., ZHENG H., WU W., YU X.: 'Probability distribution of blackouts in complex power networks'. Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2007), New Orleans, USA, May 2007, pp. 69–72

[5] TAM W.M., LAU F.C.M., TSE C.K.: 'Modeling the telephone call network'. Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2007), New Orleans, USA, May 2007, pp. 453–456

[6] TAM W.M., LAU F.C.M., TSE C.K.: 'Complex network modeling of a call network', *IEEE Trans. Circuit Syst. I*, 2009, **56**, (2), pp. 416–429

doi: 10.1049/iet-com.2008.0503

[7] WANG X.F., CHEN G.: 'Synchronization in scale-free dynamical networks: robustness and fragility', *IEEE Trans. Circuit Syst. I*, 2002, **49**, (1), pp. 154–62

[8] LI X., CHEN G.: 'Synchronization and desynchronization of complex dynamical networks: an engineering viewpoint', *IEEE Trans. Circuit Syst. I*, 2003, **50**, (11), pp. 1381–1390

[9] WU C.W.: 'Synchronization in systems coupled via complex networks'. Proc., IEEE Int. Symp. Circuits and Systems (ISCAS 2004), Vancouver, Canada, May 2004, pp. 724–727

[10] LI Z., CHEN G.: 'Global synchronization and asymptotic stability of complex dynamical networks', *IEEE Trans. Circuits Syst. II*, 2006, **53**, (1), pp. 28–33

[11] JIANG G.P., TANG W.K.S., CHEN G.: 'A state-observer-based approach for synchronization in complex dynamical networks', *IEEE Trans. Circuits Syst. I*, 2006, **53**, (12), pp. 2739–2745

[12] ERDOS P., RÉNYI A.: 'On the evolution of random graphs', *Publ. Math. Inst. Hung. Acad. Sci.*, 1959, **5**, pp. 17–60

[13] STROGATZ S.H.: 'Exploring complex networks', *Nature*, 2001, **410**, pp. 268–276

[14] COHEN R., HAVLIN S.: 'Scale-free networks are ultrasmall', *Phy. Rev. L*, 2003, **90**, p. 058701

[15] RICHARDSON T.J., SHROKROLLAHI M.A., URBANKE R.: 'Design of capacity-approaching irregular low-density parity-check codes', *IEEE Trans. Inform. Theory*, 2001, **47**, pp. 619–637

[16] RICHARDSON T.J., URBANKE R.: 'The capacity of low-density parity-check codes under message-passing decoding', *IEEE Trans. Inform. Theory*, 2001, **47**, pp. 599–618

[17] LUBY M., MITZENMACHER M., SHOKROLLAHI A., SPIELMAN D.: 'Analysis of low density codes and improved designs using irregular graphs'. Proc., ACM Symp. Theory of Computing, Dallas, USA, May 1998, pp. 249–258

[18] CHUNG S.-Y., RICHARDSON T.J., URBANKE R.: 'Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation', *IEEE Trans. Inform. Theory*, 2001, **47**, pp. 657–670

[19] TEN BRINK S.: 'Designing iterative decoding schemes with the extrinsic information transfer chart', *AEU Int. J. Electron. Commun.*, 2000, **54**, (6), pp. 389–398

[20] MEASSON C., MONTANARI A., URBANKE R.: 'Why we can not surpass capacity: the matching condition', Arxiv preprint cs.IT/0510045, 2005

[21] GUILLAUME J.-L., LATAPY M.: 'A realistic model for complex networks', http://arxiv.org/abs/cond-mat/0307095

[22] SCHLEGEL C.B., PEREZ L.C.: 'Trellis and turbo coding' (Wiley-IEEE Press, March 2004)

[23] MACKAY D.J.C.: 'Encyclopedia of sparse graph codes', http://wol.ra.phy.cam.ac.uk/mackay/codes/data.html

[24] HU X.Y., ELEFTHERIOU E., ARNOLD D.M.: 'Regular and irregular progressive edge-growth tanner graphs', *IEEE Trans. Inform. Theory*, 2005, **51**, (1), pp. 386–398

doi: 10.1049/iet-com.2008.0503