Science Discovery
2017; 5(5): 348-354
http://www.sciencepublishinggroup.com/j/sd
doi: 10.11648/j.sd.20170505.18
ISSN: 2331-0642 (Print); ISSN: 2331-0650 (Online)

Quasi-cyclic Expansion Algorithm for Protograph LDPC Codes Based on PEG and PH

Ruiyan Du[1], Qingping Zhou[2], Fulai Liu[1], Dong Wang[3]

[1]Engineer Optimization & Smart Antenna Institute, Northeastern University at Qinhuangdao, Qinhuangdao, China
[2]Tangshan Normal University, Tangshan, China
[3]Shenzhen Transsion Holdings Limited, Shenzhen, China

Email address:
ruiyandu@126.com (Ruiyan Du), 470376608@qq.com (Qingping Zhou), fulailiu@126.com (Fulai Liu), 470130228@qq.com (Dong Wang)

To cite this article:
Ruiyan Du, Qingping Zhou, Fulai Liu, Dong Wang. Quasi-cyclic Expansion Algorithm for Protograph LDPC Codes Based on PEG and PH. *Science Discovery.* Vol. 5, No. 5, 2017, pp. 348-354. doi: 10.11648/j.sd.20170505.18

Received: June 30, 2017; **Accepted**: August 4, 2017; **Published**: August 14, 2017

Abstract: One of the important code of modern coding theory, the protograph LDPC code technology has been gained more and more attention due to its low encoding complexity, fast decoding speed, low BER, and simple optimization analysis process etc. the quasi-cyclic expansion algorithm of the protograph LDPC codes, named as PQCE algorithms, can complete the extension of protograph and eventually get the protograph LDPC codes. However, the existing PQCE algorithms may be with a low convergence rate, or exist many short cycles in the check matrix. To solve the above problem, a Quasi-cyclic expansion algorithm for protograph LDPC codes based on PEG and PH is proposed in this paper, referred as PEG-PH-PQCE algorithm. In the proposed algorithm, base matrix is acquired by PEG parallel edges elimination expansion algorithm during the first-step expansion of protograph. Then, the second-step expansion is completed, in which the initial index matrix is obtained by PEG quasi-cyclic expansion algorithm, and the check matrix is acquired by using the Hill Climbing algorithm to optimizing the initial index matrix. Simulation results demonstrate the effectiveness the validity of the proposed algorithm, such as, a small number of short cycles and high convergence rate, etc.

Keywords: Protograph LDPC Codes, Quasi-Cyclic Expansion, Short Cycle

基于PEG和PH的原模图LDPC码准循环扩展算法

杜瑞燕[1]，周庆平[2]，刘福来[1]，王冬[3]

[1]工程优化设计与智能天线研究所，东北大学秦皇岛分校，秦皇岛，中国
[2]唐山师范学院，唐山，中国
[3]深圳传音控股有限公司，深圳，中国

邮箱
ruiyandu@126.com（杜瑞燕），470376608@qq.com（周庆平），fulailiu@126.com（刘福来），470130228@qq.com（王冬）

摘要：作为现代编码理论的重要编码之一，原模图LDPC （Low Density Parity Check, LDPC）码由于具有编码复杂度低、译码速度快、误码性能良好、优化分析过程简单等诸多优点，受到了越来越多的关注。原模图LDPC码的准循环扩展算法，简称为PQCE算法，能够完成原模图扩展并最终得到原模图LDPC码。但已有的PQCE算法存在收敛速度慢或其校验矩阵存在大量短环等问题，为此，本文提出了一种基于PEG（Progressive Edge Growth, PEG）和PH扩展的原模图准循环扩展算法，简称PEG-PH-PQCE算法，该算法首先采用PEG去重边扩展算法进行第一步扩展，进而通过PEG准循环扩展算法获取初始指数矩阵，最后利用登山算法（Hill Climbing, HC）对初始指数矩阵优化得校验矩阵，完成第二步扩展。仿真表明，利用所提算法构造的校验矩阵中短环数量较少，收敛速度有显著提高。

关键词：原模图 LDPC 码，准循环扩展，短环

1．引言

作为提高传输可靠性和节省频谱资源的有效技术，信道编码技术已经广泛被应用到各类通信系统以及计算机存储和运算系统中。

当前信道编码方案主要包括两大类：线性分组码和卷积码。线性分组码是发展较早的一类纠错码，并以群、环、域等相关数学理论为基础。作为线性分组码发展的重大突破，Prange于1957年提出了循环码[1]，之后又出现了两类循环码BCH码和RS码。Elias于1955年提出了卷积码[2]，相比于分组码，卷积码具有时延少、高编码增益等优点。但上述传统编码始终无法逼近香农限。

虽然Turbo码和低密度奇偶校验（Low Density Parity Check, LDPC, LDPC）码都能够逼近香农限，但与Turbo码相比，LDPC码具有诸多优势：1）LDPC码译码器具有全并行结构，可以获得更高的译码速度；2）LDPC码具有“自交织”性，省去了复杂的交织器，从而降低了传输时延；3）LDPC码具有较低的误码平层；4）高码率LDPC码不需要通过打孔，实现设计更加灵活。因此，LDPC码受到越来越多的关注。

目前LDPC码校验矩阵的构造方法分为三类：随机构造法、结构化构造法以及混合构造法。随机构造方法根据特定的设计准则和Tanner图来搜寻与确定矩阵中非零元素的位置，随机构造方法主要有渐进边增长（Progressive Edge Growth, PEG）算法[3，4]、近似外信息度算法[5]等。结构化构造法主要利用抽象代数、有限几何、图论等方法构造出具有循环或者准循环结构的校验矩阵。混合构造法[6，7]利用了随机构造和结构化构造方法的优点，主要思想是在保持局部子结构的情况下，根据约束条件对子结构进行随机置换。目前，混合构造法主要侧重于QC-LDPC码的构造。J. Thorpe于2003年提出了原模图LDPC码[8]，将对LDPC码的设计简化为对原模图的设计，研究表明[9]，原模图LDPC码最具有优势。利用原模图构造的RA码[10]、ARA码[11]等一系列码型既能利用LDPC码的迭代译码算法进行并行译码，也能利用级联码的串行高速编码算法进行快速高效编码，解决了LDPC码编码需要大量存储空间以及计算复杂度高的弊病；同时，原模图优化分析过程简单，码率和码长设计灵活并且具有码率兼容性。所以原模图LDPC码受到了越来越多的关注。而通过原模图准循环扩展构造LDPC码的算法，简称为PQCE算法，它的优劣对原模图LDPC码的性能起着至关重要的作用。2012年，R. Asvadi等人于提出了利用环提升法构造具有较低误码平层的非规则有限长度原模图LDPC码的方法[12]。Y. Wang等于2013年提出将原模图扩展为具有层次结构的QC-LDPC码的算法，使得构造的QC-LDPC码具有较大围长[13]，同年Hosung Park等提出一种基于原模图的QC-LDPC码构造算法，但该算法不可避免短环[14]。2016年，包建荣等提出了一种针对原模图校验节点扩展QC-LDPC编码的构造方法，具有较好的误比特率性能[15]。2017年，为改善空间耦合LDPC码在突发删除信道下的性能，张昭基等提出一种非对称空间耦合结构，能够明显改善SC-LDPC码的单突发删除纠错性能[16]。

考虑已有的PQCE算法存在收敛速度慢或其校验矩阵存在大量短环等问题，本文提出了一种基于PEG和PH扩展的原模图准循环扩展算法，简称PEG-PH-PQCE算法。首先采用PEG去重边扩展算法进行第一步扩展，进而通过PEG准循环扩展算法获取初始指数矩阵，最后利用登山算法（Hill Climbing, HC）对初始指数矩阵优化得校验矩阵，完成第二步扩展。仿真实验表明，利用所提算法构造的校验矩阵中短环数量较少，收敛速度有显著提高。

2．PEG-PH-PQCE算法

该算法首先通过PEG去重边算法对原模图进行扩展获得基矩阵 $\boldsymbol{H}_{\text{base}}$，然后利用PEG准循环扩展算法得到初始指数矩阵 $\boldsymbol{H}'_{\text{shift}}$，接下来经HC算法对初始指数矩阵 $\boldsymbol{H}'_{\text{shift}}$ 中的循环移位值进行优化以消除短环，最后利用循环置换子矩阵和全零子矩阵进行扩展得到校验矩阵。

2.1. PEG算法

PEG算法是一种保证LDPC码校验矩阵局部围长最大化的高效快速构造方法。在满足给定度分布的条件下，按照变量节点排列的先后顺序，采用逐边添加的方式在变量节点和校验节点之间添加边。假设当前已经通过PEG算法完成了前 $j-1$ 个变量节点的边的添加并得到了一个Tanner图，在添加第 j 个变量节点和校验节点之间的边的时候，保证每次在向现有Tanner图中添加新边后都能够使得新构成的Tanner图中变量节点 S_j 的局部围长最大化。

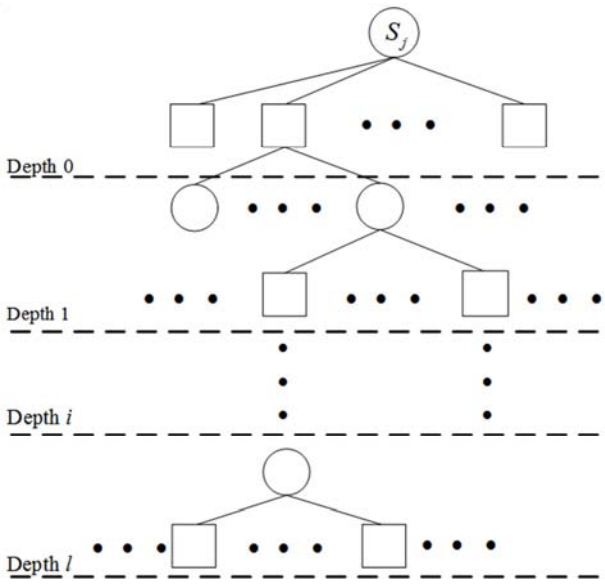

图1 沿变量节点S_j展开的深度为l的树图。

为了便于PEG算法的具体描述，首先要定义相关概念以及变量符号。

$N_{S_j}^l$：根据Tanner图从变量节点 S_j 展开成深度（Depth）为 l 的树图中包含的所有校验节点的集合；

$\overline{N}_{S_j}^l$：若 V_C 表示所有校验节点的集合，则 $\overline{N}_{S_i}^l$ 表示以 V_C 为全集 $\overline{N}_{S_j}^l$ 的补集；

E：Tanner图中所有边集合，$E = E_{S_1} \cup E_{S_2} \cup \cdots \cup E_{S_n}$，其中 E_{S_j} 表示所有与变量节点 S_j 相连的边的集合；

D_S：表示变量节点的度分布序列，$D_S = (d_{S_1}, d_{S_2}, \ldots, d_{S_n})$，其中 d_{S_j} 表示变量节点 S_j 的度，且满足不减少的特性 $d_{S_1} \le d_{S_2}, \ldots, \le d_{S_n}$；

D_C：表示校验节点的度分布序列，$D_C = (d_{C_1}, d_{C_2}, \ldots, d_{C_m})$，其中 d_{C_i} 表示校验节点 C_i 的度，且满足不减少的特性 $d_{C_1} \le d_{C_2} \le \ldots \le d_{C_m}$；

邻居节点：所有与某变量节点或者校验节点相连的校验节点或者变量节点的集合称作该节点的邻居节点。

图1表示根据Tanner图以变量节点 S_j 为根节点扩展为深度为 l 的树图，树图扩展的每一层的节点都不相同，初始节点 S_j 与在第 l 层首次出现的变量节点的距离为 $2l$，而与第 l 层首次出现的校验节点的距离为 $2l+1$。设需要通过PEG算法扩展的Tanner图有 m 个校验节点以及 n 个变量节点则PEG算法的伪代码描述如下：

For $j = 1$ *to* n

For $k = 1$ *to* d_{S_j}

If $k = 1$

$E_{S_j}^1 \leftarrow E(C_i, C_j)$，$E_{S_j}^1$ 表示连接到变量节点 S_j 的第一条边，C_i 是现有Tanner图中度最小的校验节点。

Else

将树图扩展到第 l 层，使得 $N_{S_j}^l$ 中的校验节点个数不再增加，并且满足 $N_{S_j}^l < m$ 或者 $\overline{N}_{S_j}^l \neq \varphi$ 同时 $\overline{N}_{S_j}^{l+1} = \varphi$，设置边 $E_{S_j}^k \leftarrow E(C_i, C_j)$，$E_{S_j}^k$ 表示连接到变量节点 S_j 的第 k 条边。C_i 表示 $\overline{N}_{S_j}^l$ 中所有校验节点中度最小的校验节点。

End if

End for

与第 l 层首次出现的校验节点的距离为 $2l+1$。设需要通过PEG算法扩展的Tanner图有 m 个校验节点以及 n 个变量节点则PEG算法的伪代码描述如下：

For $j = 1$ *to* n

For $k = 1$ *to* d_{S_j}

If $k = 1$

$E_{S_j}^1 \leftarrow E(C_i, C_j)$，$E_{S_j}^1$ 表示连接到变量节点 S_j 的第一条边，C_i 是现有Tanner图中度最小的校验节点。

Else

将树图扩展到第 l 层，使得 $N_{S_j}^l$ 中的校验节点个数不再增加，并且满足 $N_{S_j}^l < m$ 或者 $\overline{N}_{S_j}^l \neq \varphi$ 同时 $\overline{N}_{S_j}^{l+1} = \varphi$，设置边 $E_{S_j}^k \leftarrow E(C_i, C_j)$，$E_{S_j}^k$ 表示连接到变量节点 S_j 的第 k 条边。C_i 表示 $\overline{N}_{S_j}^l$ 中所有校验节点中度最小的校验节点。

End if

End for

End for

下面具体讨论PEG算法树图停止扩展的两个条件，根据图1沿着变量节点 S_j 扩展到深度为 l 的树图可知，需要满足以下任意一个条件：

（a）树图中校验节点的集合 $N_{S_j}^l$ 停止增长，即 $N_{S_j}^l = N_{S_j}^{l+1}$，同时该集合中元素的数目小于 m；

（b）在第 $l+1$ 扩展中将包含全部 m 个校验节点，即 $\overline{N}_{S_j}^l \neq \varphi$ 同时 $\overline{N}_{S_j}^{l+1} = \varphi$。

当条件（a）满足时，选择现有Tanner图之外的校验节点 C_i 与当前变量节点 S_j 连接，由于校验节点第一次在Tanner图中出现，并且仅与一个变量节点 S_j 相连，所以在 C_i 和 S_j 之间添加新边之后，新构成的Tanner图中不会增加任何环。当条件（b）满足时，全部 m 个校验节点都可以通过树图展开得到，连接任何一个校验节点都将会构成环，所以必须要保证构成的环的环长最大。通过条件（b）选择的校验节点必然在树图的第 $l+1$ 层中，这样就可以保证构成的环的环长为 $2(l+2)$，进而保证了当前变量节点的局部围长最大化。

2.2. 去重边扩展

由上文分析可知Tanner图中的短环会对LDPC码的性能产生影响，如果存在重边则围长仅仅为2，所以必须首先将原模图中的重边去除。第一步扩展过程的基本思想是首先将原模图扩展 L 倍，L 必须不小于最大重边数，然后进行边置换操作，达到除去原模图中重边的目的。为了保证原模图的特性在扩展之后得到继承，边置换操作只能在同一变量节点或者校验节点扩展出来的 L 个节点之间进行。由于 L 大于最大重边数，所以通过适当的边置换操作即可保证扩展之后的Tanner图中不存在重边。扩展中一般采用PEG算法，但需要在算法中加入与原模图相关的约束条件。

设原模图基础矩阵维数为 $m \times n$，则第一步扩展之后得到的基矩阵 $\boldsymbol{H}_{\text{base}}$ 的维数为 $mL \times nL$，其中 L 为第一步扩展次数。设 S_j 为扩展之后 $\boldsymbol{H}_{\text{base}}$ 中的变量节点，该变量节点由原模图中变量节点 $S_{proto}^{j'}$ 扩展得到，其中 $(j'-1)L < j \le j'L$，$1 < j' \le n$。在沿 S_j 扩展树图时，所有可以与 S_j 之间添加边的校验节点的集合记为 C_{allow}，C_{allow} 需要考虑到与原模图中变量节点和校验节点的度同时对应。将校验节点分为 m 组 $C = (C_{\text{set}}^1, C_{\text{set}}^2, \ldots, C_{\text{set}}^m)$，若校验节点 C_i 包含在子集 $C_{sub}^{i'}$ 中，则有 $(i'-1)L < i \le i'L$，其中 $1 < i' \le m$。每一子

集都有该子集中校验节点当前所能容纳边的最大数目，即 $EN_j=(EN_j^1,EN_j^2,\ldots,EN_j^m)$， $EN_j^{i'}$ 的初始值同原模图基础矩阵元素 $h_{proto}(i',j')$ 的值相同，即 $EN_j^{i'}=h_{proto}(i',j')$。根据2.1部分的PEG算法，PEG去重边扩展算法描述如下：

For $j=1$ *to* n

For $j=1$ *to* L

For $k=1$ *to* d_{S_j}

If $k=1$

$E_{S_j}^1\leftarrow E(C_i,C_j)$， $E_{S_j}^1$ 表示连接到变量节点 S_j 的第一条边。在由已添加边 $E_{S_1}\cup E_{S_2}\cup\cdots\cup E_{S_{j-1}}$ 组成的现有Tanner图条件下， C_i 是 C_{allow} 中度最

小的校验节点，更新 C_{allow} 中元素。 $EN_j^{i'}=EN_j^{i'}-1$，若 $EN_j^{i'}=0$ 则不能在校验节点集合 $N_{sub}^{i'}$ 与 S_j 之间添加边，更新 C_{allow}。

Else

将树图扩展到第 l 层，使得 $N_{S_j}^l$ 中的校验节点个数不再增加，并且满足 $N_{S_j}^l<m$ 或者 $\overline{N}_{S_j}^l\cap C_{allow}\neq\varphi$ 同时 $\overline{N}_{S_j}^{l+1}\cap C_{allow}=\varphi$，设置边 $E_{S_j}^k\leftarrow E(C_i,C_j)$， $E_{S_j}^k$ 表示连接到变量节点 S_j 的第 k 条边。C_i 表示 $\overline{N}_{S_j}^l\cap C_{allow}$ 中所有校验节点中度最小的校验节点。更新 C_{allow} 中元素。 $EN_j^{i'}=EN_j^{i'}-1$，若 $EN_j^{i'}=0$ 则不能在校验节点集合 $C_{sub}^{i'}$ 与 S_j 之间添加边，更新 C_{allow} 中元素。

End if

End for

End for

End for

2.3．准循环扩展

通过PEG去重边扩展算法得到基矩阵 $\boldsymbol{H}_{\text{base}}$ 之后，需要利用PH准循环扩展算法将 $\boldsymbol{H}_{\text{base}}$ 扩展为准循环校验矩阵 $\boldsymbol{H}$。在基矩阵 $\boldsymbol{H}_{\text{base}}$ 中元素“0”对应校验矩阵 $\boldsymbol{H}$ 中的全零子矩阵，元素“1”对应 $\boldsymbol{H}$ 中的循环置换子矩阵。首先利用PEG准循环扩展算法得到初始指数矩阵 $\boldsymbol{H}'_{\text{shift}}$，然后通过HC算法对 $\boldsymbol{H}'_{\text{shift}}$ 中的循环移位值进行优化，以消除 $\boldsymbol{H}'_{\text{shift}}$ 中的短环获得最终的指数矩阵 $\boldsymbol{H}_{\text{shift}}$，最后利用循环置换子矩阵和全零子矩阵对 $\boldsymbol{H}_{\text{shift}}$ 扩展得到校验矩阵 $\boldsymbol{H}$。首先讨论通过PEG准循环扩展算法得到 $\boldsymbol{H}'_{\text{shift}}$ 的过程。

（1）利用PEG准循环扩展算法得到初始指数矩阵

设基矩阵 $\boldsymbol{H}_{\text{base}}$ 的维数为 $m'\times n'$，每一个循环置换子矩阵的维数为 $p\times p$。根据循环矩阵的性质，利用PEG算法进行准循环扩展获得 $\boldsymbol{H}$ 时，只需要通过确定每个循环置换子矩阵第一列中“1”的位置，即可确定循环置换子矩阵中剩余“1”元素的位置，同时确定了该循环置换子矩阵的循环移位值。

将校验矩阵 $\boldsymbol{H}$ 按列平均分为 n' 部分，则有 $\boldsymbol{H}=[\boldsymbol{H}_1\quad\boldsymbol{H}_2\quad\cdots\quad\boldsymbol{H}_{n'}]$。由此PEG算法展开时，以每一个子矩阵第一列对应的变量节点 S_j 为根节点展开为树图，其中 $j=1,p+1,\ldots,(n'-1)p+1$， p 表示循环置换子矩阵的维数同时也是第二步扩展次数。沿 S_j 展开为树图时，所有能够同 S_j 通过边连接的校验节点组成的集合表示为 C_{allow}，C_{allow} 用来限制扩展得到的准循环校验矩阵满足约束条件，IPEG准循环扩展算法具体算法描述如下所示：

For $j=1$ *to* $(n'-1)p+1$ *step* p

For $k=1$ *to* d_{S_j}

If $k=1$

$E_{S_j}^1\leftarrow E(C_i,C_j)$， $E_{S_j}^1$ 表示连接到变量节点 S_j 的第一条边，在由已添加边 $E_{S_1}\cup E_{S_2}\cup\cdots\cup E_{S_{j-1}}$ 组成的现有Tanner图条件下， C_i 是 C_{allow} 中度最小的校验节点。根据 $E_{S_j}^1$ 的位置生成循环置换子矩阵，添加到最终的校验矩阵中，更新 C_{allow} 中元素。

Else

将树图扩展到第 l 层，使得 $N_{S_j}^l$ 中的校验节点个数不再增加，并且满足 $N_{S_j}^l<m$ 或者 $\overline{N}_{S_j}^l\cap C_{\text{allow}}\neq\varphi$ 同时 $\overline{N}_{S_j}^{l+1}\cap C_{allow}=\varphi$，设置边 $E_{S_j}^k\leftarrow E(C_i,C_j)$， $E_{S_j}^k$ 表示连接到变量节点 S_j 的第 k 条边。 C_i 表示 $\overline{N}_{S_j}^l\cap C_{allow}$

中所有校验节点中度最小的校验节点。根据 $E_{S_j}^k$ 位置生成循环置换子矩阵添加到最终的校验矩阵中，更新 C_{allow} 中元素。

End if

End for

End for

PEG准循环扩展算法可以直接获得校验矩阵 $\boldsymbol{H}$，但是 $\boldsymbol{H}$ 中存在较多的短环。为了能够通过HC算法进一步优化，在PH算法中仅仅利用PEG准循环扩展算法得到初始指数矩阵 $\boldsymbol{H}'_{\text{shift}}$，这就需要根据每次添加边时选择的校验节点得到循环移位值。然后通过HC算法对 $\boldsymbol{H}'_{\text{shift}}$ 中的循环移位值进一步优化得到最终的指数矩阵 $\boldsymbol{H}_{\text{shift}}$。

（2）HC算法优化初始指数矩阵

HC算法是通过对基矩阵扩展构造QC-LDPC码的常用算法，文献[17]中给出了其详细的算法步骤。原算法中对指数矩阵中每一个循环移位值随机赋值，导致算法收敛速度慢。在PH准循环扩展算法中，直接利用PEG准循环扩展算法得到的初始指数矩阵作为初值，然后在每次迭代中按照一定规则改变指数矩阵中的循环移位值，最后达到减少校验矩阵中短环数量的目的。

令 $2g$ 表示需要优化的短环的最大环长，首先需要确定基矩阵 $\boldsymbol{H}_{\text{base}}$ 中所有环长小于 $2g$ 的环。对于较小的原模图而言，可以对具有较大环长的环进行优化，但是随着原模图中节点或者边数量的增多，会导致基矩阵中环数量过多，优化过程复杂，所以需要合理选择 $2g$ 的大小。 $\boldsymbol{\Gamma}^l$ 表示由基矩阵 $\boldsymbol{H}_{\text{base}}$ 中所有环长为 $2l$ 的环中的节点组成的矩阵，则 $\boldsymbol{\Gamma}^l$ 可表示为：

$$\boldsymbol{\Gamma}^{l}=\begin{bmatrix} C_{1,1} & S_{1,2} & \cdots & S_{1,2l} \\ C_{2,1} & S_{2,2} & \cdots & S_{2,2l} \\ \vdots & \vdots & \ddots & \vdots \\ C_{num_l,1} & S_{num_l,2} & \cdots & S_{num_l,2l} \end{bmatrix}_{num_l\times 2l} \tag{1}$$

其中，$\boldsymbol{H}_{\text{base}}$ 的第 t 条环中的第 n_1 个节点为校验节点 C_{t,n_1}，其中 $n_1=1,3,\ldots,2l-1$，$\boldsymbol{H}_{\text{base}}$ 的第 t 条环中第 n_2 个节点为变量节点 S_{t,n_2}，其中 $n_2=2,4,\ldots,2l$。num_l 表示 $\boldsymbol{H}_{\text{base}}$ 中环长为 $2l$ 的环的总数量。根据 $\boldsymbol{H}_{\text{base}}$ 和 $\boldsymbol{H}_{\text{shift}}$ 之间的关系，环可以通过每个循环置换子矩阵的循环移位值表示，则可根据 $\boldsymbol{\Gamma}^{l}$ 得到：

$$\boldsymbol{\Gamma}^{l}_{\text{shift}}=\begin{bmatrix} a(C_{1,1},S_{1,2}) & a(C_{1,3},S_{1,2}) & \cdots & a(C_{1,2l-1},S_{1,2l}) \\ a(C_{2,1},S_{1,2}) & a(C_{2,3},S_{2,2}) & \cdots & a(C_{2,2l-1},S_{2,2l}) \\ \vdots & \vdots & \ddots & \cdots \\ a(C_{num_l,1},S_{num_l,2}) & a(C_{num_l,3},S_{num_l,2}) & \cdots & a(C_{num_l,2l-1},S_{num_l,2l}) \end{bmatrix} \tag{2}$$

由文献[18]可知，循环置换矩阵链 $\boldsymbol{P}^{a(i_{t,1},j_{t,1})}\to\cdots\to\boldsymbol{P}^{a(i_{t,k},j_{t,k})}\to\cdots\to\boldsymbol{P}^{a(i_{t,2l},j_{t,2l})}$ 构成长度为 $2l$ 的环的充要条件为：

$$\sum_{k=1}^{2l}(-1)^{k-1}a(i_{t,k},j_{t,k})\equiv 0 \mod p \tag{3}$$

该循环置换矩阵链中子矩阵移位值与 $\boldsymbol{\Gamma}^{l}_{\text{shift}}$ 中第 t 行中元素对应，其中 $t=1,2,\cdots,num_l$。通过调整路径元素的循环移位值使得（3）式不再成立，进而达到消除或者减少环长为 $2l$ 的环的目的。

对于全部 (i,j)，由0到 $p-1$ 顺次改变某一个循环置换子矩阵 $\boldsymbol{P}(i,j)$ 的循环移位值 $a(i,j)$，由（3）式计算 $\boldsymbol{P}(i,j)$ 在具有不同循环移位值 q 时，在不同长度的环分别中出现的总次数，并通过 $num_l^q(i,j)$ 表示。由于环长长短不同对LDPC码性能的影响也不同，所以在进行优化的时候需要对不同长度的环的个数添加不同的权重值 w_l。由于环长越短的环对LDPC码性能影响越大，故环长越小的环对应的权重值 w_l 越大。定义当 $\boldsymbol{P}(i,j)$ 的循环移位值为 q 时的代价值 $\boldsymbol{\Lambda}(i,j,q)$ 为：

$$\Lambda(i,j,q)=num_2^q(i,j)w_2+num_3^q(i,j)w_3+\cdots+num_g^q(i,j)w_g \tag{4}$$

$\boldsymbol{\Lambda}(i,j,q)$ 越大说明当 $\boldsymbol{P}(i,j)$ 的循环移位值选择 q 时LDPC码性能越差。HC算法具体步骤如下：

step1 初始化，由PEG准循环扩展算法得到初始的指数矩阵 $\boldsymbol{H}'_{\text{shift}}$；

step2 通过 $\boldsymbol{H}_{\text{shift}}$ 获得 $\boldsymbol{H}_{\text{base}}$ 中环的循环移位值表示 $\boldsymbol{\Gamma}^{2}_{\text{shift}},\boldsymbol{\Gamma}^{3}_{\text{shift}},\cdots,\boldsymbol{\Gamma}^{g}_{\text{shift}}$；

step3 根据式（3）计算获得代价值矩阵 $\boldsymbol{\Lambda}_{m\times n\times q}$；

step4 对于任意的 (i,j) 计算使得 $\boldsymbol{\Lambda}_{i,j,q}$ 最小的循环移位值 $\tilde{q}_{i,j}$ 以及对应的最小代价值，原始代价值可表示为 $\overline{\boldsymbol{\Lambda}}_{i,j}$

$$\tilde{q}_{i,j}=\arg\min_{q:0\le q\le p-1}\Lambda_{i,j,q}$$

$$\tilde{\Lambda}_{i,j}=\min_{q:0\le q\le p-1}\Lambda_{i,j,q}$$

step5 计算使得代价值下降最多的元素位置 $i_{\max}$ 和 $j_{\max}$：

$$i_{\max},j_{\max}=\arg\max_{i:1\le i\le m,j:1\le j\le n}\widetilde{\boldsymbol{\Lambda}}_{i,j}-\overline{\boldsymbol{\Lambda}}_{i,j}$$

step6 如果 $\widetilde{\boldsymbol{\Lambda}}_{i,j}-\overline{\boldsymbol{\Lambda}}_{i,j}>0$，将指数矩阵 $\boldsymbol{H}_{\text{shift}}$ 中元素 $P(i_{\max},j_{\max})$ 更新为 $\tilde{q}(i_{\max},j_{\max})$，返回。

第二步继续优化；如果 $\widetilde{\boldsymbol{\Lambda}}_{i,j}-\overline{\boldsymbol{\Lambda}}_{i,j}=0$，表明已经无法继续优化迭代终止。

由HC算法得到最终的指数矩阵 $\boldsymbol{H}_{\text{shift}}$ 后，利用循环置换子矩阵和全零子矩阵对其扩展即可得到最终的校验矩阵。

3. 仿真结果与分析

为了验证PEG-PH-PQCE算法的有效性，我们在matlab平台上对PEG-PEG-PQCE算法、PEG-HC-PQCE算法以及PEG-PH-PQCE算法三种原模图扩展算法进行仿真实验，并分别从短环数量、算法运行时间以及误码性能角度对算法进行性能分析。

首先对PEG-PH-PQCE算法和PEG-HC-PQCE算法的算法运行时间，算法迭代次数以及短环个数进行比较，分别利用两种扩展算法对原模图 G_1 进行扩展，得到两次运行结果数据并记录，如表1所示。

分析比较表1数据，两种算法扩展得到的 $\boldsymbol{H}_{\text{shift}}$ 中的4个数相同并且都为0，6个数也十分相近并且没有确定的大小关系，但是PEG-HC-PQCE算法的优化时间以及算法迭代次数远远多于PEG-PH-PQCE算法。故可得出结论：相比于PEG-PEG-PQCE算法，PEG-PH-PQCE算法大大减少了校验矩阵中短环个数，相比于PEG-HC-PQCE算法，所提算法的收敛速度提高了4倍。

表1 PEG-PH-PQCE和PEG-HC-PQCE算法比较。

PQCE算法	PEG-PH		PEG-HC	
仿真次数	1	2	1	2
HC算法运行前H_{shift}中6环个数	781	808	834	860
HC算法运行前H_{shift}中4环个数	0	0	42	31
HC算法运行后H_{shift}中6环个数	477	470	471	480
HC算法运行后H_{shift}中4环个数	0	0	0	0
HC算法迭代次数	69	44	296	275
算法所需时间（s）	204.28028	228.02659	878.78842	900.06389

下面对PEG-PEG-PQCE算法、PEG-HC-PQCE算法以及PEG-PH-PQCE算法三种扩展算法的误码性能进行仿真分析。仿真参数设置为：第一步扩展次数 $\text{lift1}=4$，第二步扩展次数 $\text{lift2}=32$，最大译码迭代次数 $\text{iter}=20$，调制

方式采用BPSK调制方式，BP译码算法，信道为AWGN信道。对原模图 G_1 和 G_2 分别利用上述三种扩展算法进行扩展得到的原模图LDPC码的误码性能曲线，如图2和图3。在信噪较小时误码率曲线接近，但是随着信噪比增加，可以发现三种算法构造的LDPC码的误码率满足关系：$BER_{PEG\text{-}PEG}$ 的误码率最高，而 $BER_{PEG\text{-}PH}$ 与 $BER_{PEG\text{-}HC}$ 则比较接近并且要低于 $BER_{PEG\text{-}PEG}$。相比于PEG-PEG-PQCE算法PEG-PH-PQCE算法有效降低了所构造原模图LDPC码的误码率，验证了算法的有效性。

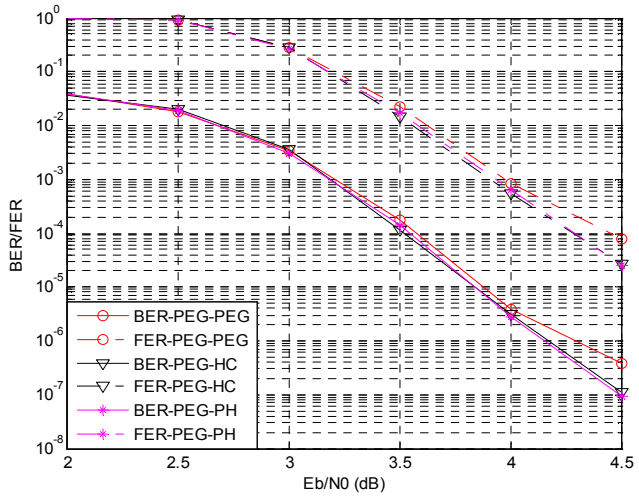

图2 原模图为G_1时相关算法性能曲线。

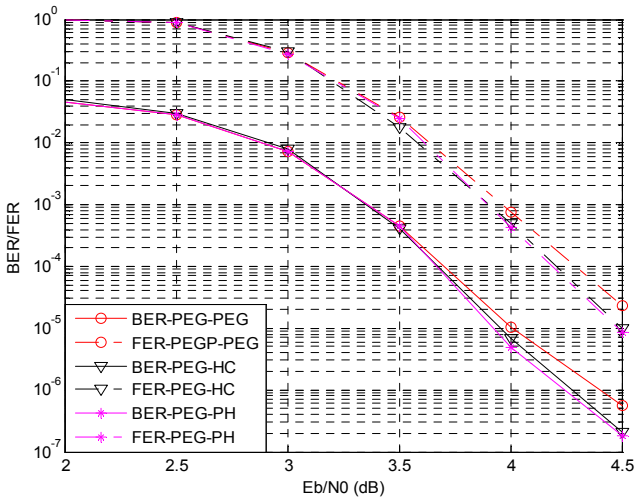

图3 原模图为G_2时相关算法性能曲线。

4．结论

针对现有的PQCE算法构造的校验矩阵中存在大量短环和较慢的收敛速度的问题，提出了PEG-PH-PQCE算法。该算法首先利用PEG去重边扩展算法获得基矩阵完成原模图第一步扩展，然后通过PEG准循环扩展算法得到初始指数矩阵，最后利用登山算法对初始指数矩阵进一步优化得到校验矩阵。仿真实验首先分析了影响原模图LDPC码性能的仿真参数，然后分别从短环数量、算法运行时间、迭代次数以及所构造原模图LDPC码的误码性能验证了PEG-PH-PQCE算法的有效性。

致谢

本文成果获得如下项目支持：新世纪优秀人才支持计划项目（NCET-13-0105）；河北省高校百名优秀创新人才支持计划项目（BR2-259）；河北省自然科学基金项目（No. F2016501139）；中国高等教育博士研究生专项科研基金（No.20130042110003）；中央高校基本科研业务费专项资金资助项目（No. N142302001，No.N162304002）。

参考文献

[1] E. Prange. Cyclic error-correcting codes in two symbols [J], Air Force Cambridge Res. Center, Cambridge, MA, Tech. Note AFCRC-TN-57-103, 1957.

[2] P. Elias. Coding for noisy channels [J], IRE Convention Record, 1955, 3(4): 37-46.

[3] X. Y. Hu, E. Eleftheriou, D. M. Arnold. Progressive edge-growth Tanner graphs [C], Proc. IEEE Global Telecommun. Conf., San Antonio, TX, USA, 2001, 2: 995-1001.

[4] X. Y. Hu, E. Eleftheriou. D. M. Arnold. Regular and irregular progressive edge-growth tanner graphs [J], IEEE Trans. Inform. Theory, 2005, 51(1): 386-398.

[5] T. Tian, C. R. Jones, J. D. Villasenor, R. D. Wesel. Selective avoidance of cycles in irregular LDPC code construction [J], IEEE Trans. Commun., 2004, 52(8): 1242-1247.

[6] Nenad M, Fossorier M P C. Systematic Recursive Construction of LDPC Codes [J]. IEEE. Comm. Lett, 2004, vol.8:302-304.

[7] Andrews K, Dolinar S, Divsalar D, Jhorpe J. Design of Low-Density Parity-Check(LDPC) Codes for Deep-Space Applications [R]. IPN Progress Report 42-159, 2004, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA,USA

[8] Thorpe J. Low-Density Parity-Check (LDPC) Codes Constructed from Protographs [R]. IPN Progress Report 42-154, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, 2004.

[9] E. Prange. Cyclic error-correcting codes in two symbols [J], Air Force Cambridge Res. Center, Cambridge, MA, Tech. Note AFCRC-TN-57-103, 1957.

[10] D. Divsalar, S. Dolinar, and C. Jones. Low-rate LDPC codes with simple protograph structure [C], Proc. IEEE Int. Symp. on Inform. Theory, Adelaide, Australia, 2005, 1622-1626.

[11] Abbasfar A, Divsalar D, Yao K. Accumulate-Repeat-Accumulate Codes [J]. IEEE Trans. on Communications, 2007, 55(4):692-702.

[12] R. Asvadi, A. H. Banihashemi, M. Ahmadian-Attari. Design of finite-length irregular protograph codes with low error floors over the binary-input AWGN channel using cyclic liftings [J], IEEE Trans. Commun., 2012, 60(4): 902-907.

[13] Yige Wang, Stark C. Draper, et al. Hierarchical and High-Girth QC LDPC Codes [J], IEEE Trans. Inform. Theory, 2013, 59(7): 4553-4583.

[14] Hosung Park, Seokbeom Hong, et al. Design of Multiple-Edge Protographs for QC LDPC Codes Avoiding Short Inevitable Cycles [J], IEEE Trans. Inform. Theory, 2013, 59(7): 4598-4614.

[15] 包建荣，高西奇，刘超，姜斌．扩展原模图ＬＤＰＣ短码的优化构造[J]．华中科技大学学报(自然科学版)，44(5):35-40。

[16] 张昭基，李颖．适用于突发删除信道的非对称空间耦合LDPC码[J]．西安电子科技大学学报(自然科学版)，44(5):1-6。

[17] T. J. Richardson, R. L. Urbanke, Efficient encoding of low-density parity check codes [J]. IEEE Trans. Commun., Feb. 2001, 47: 808-821.

[18] 刘星成，程浩辉．基于PEG算法的准循环LDPC码构造方法研究[J]．电路与系统学报，8, 2009, 14(4)。