**编码构造代码说明**

目录

[一、阶梯形结构生成 1](#_Toc79483660)

[二、移位值构造 1](#_Toc79483661)

[三、掩模 2](#_Toc79483662)

[3.1针对trapping set的掩模 2](#_Toc79483663)

[3.2针对环6的掩模 3](#_Toc79483664)

# 一、阶梯形结构生成

主程序：pg\_cons.m

运行后将会调用func\_gen\_mask\_hold\_new函数，生成阶梯形结构的初步掩模矩阵，并将结果保存到pg\_xxx.txt和hold\_idx\_xxx.txt中。其中pg包含了QC-LDPC非0子矩阵的位置信息。由于实际使用的矩阵列数并不是行数的整数倍，所以无法满足完美的阶梯形，需要在少数列中保持非0子矩阵所在行的位置，这些列的编号（从1开始）保存在hold\_idx中。

func\_gen\_mask\_hold\_new函数的参数：

* base\_size：[基础矩阵的行数, 列数]
* vdeg：掩模前的列重

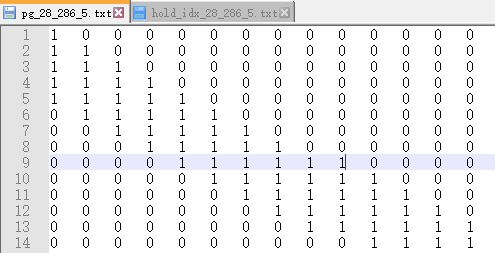


图1-1 pg\_xxx.txt矩阵示例

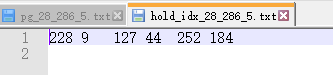


图1-2 hold\_idx\_xxx.txt矩阵示例

# 二、移位值构造

主程序：qc\_cons.cpp

运行后将会读取上一步生成的pg\_xxx.txt文件中的数据，调用cons\_qc\_pcoa函数对移位值进行构造，并将构造得到的结果保存到qc\_pcoa\_xxx.txt中。

基本原理：本算法先从随机构造的矩阵出发，搜索矩阵中所有的环4和环6，修改包含环4和环6最多的元素，使经过该元素的环4和环6数量减少，不断重复该过程，直到经过所有元素的环4和环6数量都无法减少为止。对于本项目所给定的矩阵参数，该算法可以完全消除环4，并且尽可能最小化环6的数量。

矩阵相关参数：定义在common.h中

* BASE\_ROW：基础矩阵行数
* BASE\_COL：基础矩阵列数
* BLK\_SIZE：子矩阵大小

cons\_qc\_pcoa函数的参数：

* pg：二维数组，由上一步产生，1表示该元素需要放入移位值，0表示该元素为-1（对应0子矩阵）
* mat：二维数组，构造得到的基础矩阵将会存入该数组
* tmax：进行元素优化的次数，越大计算越慢但效果越好，1000即可接近最优，10000以上基本没有明显区别，0代表不限制
  + 对于本项目的参数，设置10000可以在半分钟内完成计算
  + 如果在达到该次数之前就已经无法继续进行优化，程序会自动提前跳出

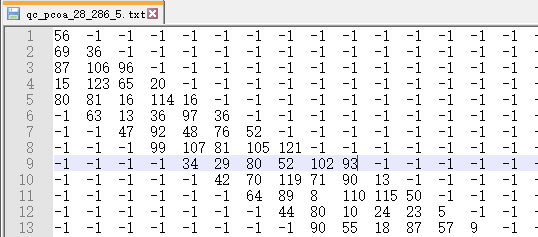


图 2-1 qc\_pcoa\_xxx.txt矩阵示例

# 三、掩模

## 3.1针对trapping set的掩模

主程序：qc\_mask.m

运行后会读取上一步生成的qc\_pcoa\_xxx.txt文件，调用func\_trapping\_set\_c1c2函数进行掩模，并将掩模结果保存到base\_matrix\_masked\_xxx.txt中。该方法能尽可能减少特定结构的trapping set的数量，避免error floor过早出现。

func\_trapping\_set\_c1c2函数的参数：

* base\_matrix：掩模前的基础矩阵，来自上一步构造的结果
* blk\_len：子矩阵大小
* max\_iter：最大掩模个数，程序将会每次掩模一个元素，每掩模一个元素都会保存一次结果，如果已经没有trapping set则会提前跳出
* min\_vdeg：最小列重，掩模过程不会使得列重小于该值

注：

* 该方法计算量较大，运行过程非常缓慢，对于本项目的参数，将列重为5掩模到4.3需要数天的时间，将列重为6掩模到4.3需要超过1周的时间，因此我们实际采用了3.2中的方法来对列重为6的构造结果进行掩模
* 该方法以及3.2的方法并不保证得到的结果可逆，但是会控制右侧方阵的对角线不被掩模来确保能方便地找到可逆的矩阵
* 该步骤得到的结果可能需要进一步掩模少量元素来实现可逆，但对最终性能影响不明显

## 3.2针对环6的掩模

主程序：qc\_mask.cpp

运行后会读取上一步生成的qc\_pcoa\_xxx.txt文件，调用mask\_one\_main函数进行掩模，并将掩模结果保存到qc\_m6\_xxx.txt中。该方法速度较快，方便对很大的矩阵进行掩模，能够尽可能减少环4（如果有的话）和环6的数量，间接减少trapping set的数量。

mask\_one\_main函数的参数：

* mat：掩模前的基础矩阵，将直接修改该矩阵进行掩模
* 该函数每次掩模1个元素，返回是否进行了有效掩模