一种基于高斯近似的极化码打孔算法

李世宝*① 高 迅① 董振威① 刘建航② 崔学荣①
①(中国石油大学(华东)海洋与空间信息学院 青岛 266580)
②(中国石油大学(华东)计算机科学与技术学院 青岛 266580)

摘 要：现有的极化码打孔算法均未考虑信道构造过程对算法性能的影响，针对这一问题，该文提出一种基于高斯近似的极化码打孔算法(GAPPC)。首先将高斯近似作为极化码构造算法，分析高斯近似与打孔算法的关系，以降低信道构造输出值为目标，引入高斯修正因子，推导出改进的高斯近似函数。然后将改进的高斯近似函数引入信道构造，对极化子信道进行排序获得信道可靠性排序集合。最后依据信道容量关系确定映射规则，选出打孔比特集合和冻结比特集合，完成打孔极化码的构建。实验结果显示，在不同的码长和码率下，误帧率和误码率均获得显著降低。

关键词：极化码；速率兼容；打孔；高斯近似
中图分类号：TN911.22 **文献标识码**：A **文章编号**：1009-5896(2021)11-3149-07
DOI: 10.11999/JEIT201007

A Puncturing Algorithm of Polar Code Based on Gaussian Approximation

LI Shibao① GAO Xun① DONG Zhenwei① LIU Jianhang② CUI Xuerong①
①(*College of Oceanography and Space Informatics*, *China University of Petroleum* (*East China*), *Qingdao* 266580, *China*)
②(*College of Computer Science and Technology*, *China University of Petroleum* (*East China*), *Qingdao* 266580, *China*)

Abstract: The influence of channel construction process on the algorithm performance is not considered in the existing polar code puncturing algorithms. To solve this problem, a Puncturing algorithm of Polar Code based on Gaussian Approximation (GAPPC) is proposed. Firstly, using Gaussian approximation for channel construction of polar code and analyzing the relationship between Gaussian approximation and puncturing algorithm, the modified Gaussian approximation function is derived to reduce the output value of channel construction with introduced Gaussian correction factors. Then the ordered channel reliability set is obtained by ordering the polarization subchannels under the channel construction with the modified Gaussian approximation function. Finally, the mapping rule is determined according to the relationship of channel capacity, and the puncturing bit set and frozen bit set are selected so that the puncturing polar code is completed. Experimental results show that the frame error rate and bit error rate are significantly reduced under different code lengths and bit rates.

Key words: Polar code; Rate-compatible; Puncturing; Gaussian Approximation(GA)

1 引言

极化码是目前已知唯一的一种被严格证明达到信道容量的信道编码方法[1]，但是由于极化码编码器是基于克罗内克积生成的[1–3]，极化码的长度总是被限制为2^n，在实际应用中，传输码字的长度不一定都是2^n，经常出现可变码长的实际需求。打孔算法是构造码长可变和码率灵活极化码的重要途径，近年来获得了研究者的广泛关注。

文献[4]首次提出极化码打孔算法，包括随机打孔和停止树打孔两种基本打孔算法，满足了码长可变的要求。文献[5]提出了一种基于删除极化矩阵的打孔算法，通过删除分别对应于打孔位和冻结位的列和行之后分析简化的极化矩阵，相对于随机打孔

收稿日期：2020-11-30；改回日期：2021-06-11；网络出版：2021-06-24
*通信作者： 李世宝 lishibao@upc.edu.cn
基金项目：国家重点研发计划(2017YFC1405203)，国家自然科学基金(61972417, 61902431, 91938204)，中央高校基本科研业务费专项资金(19CX05003A-4)
Foundation Items: The National Key R&D Program of China (2017YFC1405203), The National Natural Science Foundation of China (61972417, 61902431, 91938204), The Fundamental Research Funds for the Central Universities (19CX05003A-4)

算法可以获得1.0～5.0 dB的性能增益。文献[6]提出了准均匀打孔方案，通过比特倒置排序使得打孔比特准均匀分布，操作简单且具有较好的译码性能。文献[7]在文献[6]的基础上，提出一种倒置准均匀打孔方案，在高码率下获得更好的性能。文献[8]基于比特倒置策略和前向序列打孔提出一种新的打孔算法，提升了不同码率下的打孔性能。文献[9]提出一种适用于乘积极化码的打孔算法，性能相对于先前打孔的乘积极化码和单极性码更优。文献[10]提出并验证了使用二进制控制可以确定极化码的打孔比特集合。文献[11]结合码字重复技术提出分区打孔的思路，获取了一个更有效的信息比特集合。文献[12]将里德-所罗门(Reed-Solomon, RS)码作为极化码的外码，提出了一种平均分布打孔算法，构造了一种RS-极化码打孔方案，扩展了打孔极化码的应用范围。文献[13]提出了一种在打孔之后使用高斯近似(Gaussian Approximation, GA)对子信道进行重构的打孔算法，进一步提升了打孔算法的性能。上述的打孔算法均需要在打孔之后进行重新构造，但是重构使得算法复杂度增加。针对这一问题，文献[14]提出了一种低复杂度的打孔(Low-Complexity Puncturing, LCP)算法，在极化码构造一次的情况下使用了准均匀的打孔策略进行打孔。文献[15]提出了一种最差质量打孔(Worst-Quality Puncturing, WQP)算法，在固定信息集合下对最差质量信道进行打孔从而获取更好的打孔性能。

现有算法没有考虑信道构造环节对极化码打孔性能的影响，限制了极化码打孔性能的进一步提升。本文从信道构造出发，联合考虑打孔的特点，提出一种基于改进高斯近似的极化码打孔(Puncturing Polar Code based on Gaussian Approximation, GAPPC)算法。

2 极化码

对于二进制输入删除无记忆信道：$X \mapsto Y$，其中$X=\{0,1\}$表示二进制输入，Y是信道输出，定义信道转移概率为：$W(x|y), x\in X, y\in Y$。在N位二进制输入独立信道W进行了信道合并和分裂之后，获得N位具有相互联系的比特信道$W_N^{(i)}$，其中$i=1,2,\cdots,N, N=2^n$，子信道的容量可以表示为$I\left(W_N^{(i)}\right)$。信道的可靠性通过极化码的构造算法计算，主流的极化码构造算法[16,17]中，高斯近似构造法是最常用的构造算法[16]。在极化码构造之后选取可靠性高的比特作为信息比特集合A，而其余比特作为冻结比特集合A^C。对于二进制对称信道，冻结比特的值通常固定为0。

$\boldsymbol{G}_N$表示极化码的生成矩阵，并有$\boldsymbol{G}_N=\boldsymbol{B}_N\boldsymbol{F}^{\otimes n}$，其中$N=2^n$是极化码码长，$\boldsymbol{B}_N$是比特倒置排序矩阵，$\boldsymbol{F}^{\otimes n}$是$n$次克罗内克积，$\boldsymbol{F}\triangleq\begin{bmatrix}1&0\\1&1\end{bmatrix}$。给定$\boldsymbol{u}_1^N=(u_{\boldsymbol{A}},u_{\boldsymbol{A}^C})$作为编码器输入码字序列，极化码的编码过程可以表示为：$\boldsymbol{x}_1^N=\boldsymbol{u}_1^N\boldsymbol{G}_N$，$\boldsymbol{x}_1^N=(x_1,x_2,\cdots,x_N)$为编码后码字。$\boldsymbol{y}_1^N=(y_1,y_2,\cdots,y_N)$为译码器接收码字。

在GA构造算法中，GA构造计算如式(1)所示

$$E\left(L_N^{(i)}\right)=\begin{cases}\phi^{-1}\left(1-\left(1-\phi\left(E\left(L_{N/2}^{(i+1)/2}\right)\right)\right)^2\right),\\ \qquad i \bmod (2)=1\\ 2E\left(L_{N/2}^{i/2}\right), i \bmod (2)=0\end{cases}\tag{1}$$

第i比特u_i的对数似然值(Log Like-hood Ratio, LLR)表示为$L_N^{(i)}$，$L_N^{(i)}$满足高斯分布$L_N^{(i)}\sim N\left(\frac{2}{\sigma^2},\frac{4}{\sigma^2}\right)$，其中$\sigma^2$表示信号$u_i$的加性高斯白噪声(Additive White Gaussian Noise, AWGN)信道的方差，$E\left(L_N^{(i)}\right)$表示第i比特的LLR的密度分布函数的均值。

GA构造是蝶形结构计算的迭代过程，其中信道的$E\left(L_N^{(i)}\right)$也是蝶形计算不断迭代计算得出的。基础的蝶形计算结构如图1所示，考虑两个输入信道的差异[13]，令x_1,x_2表示为蝶形计算的输入值，y和z为其两个输出值，则蝶形计算表示如式(2)

$$\left.\begin{aligned}y&=\phi^{-1}\left(1-\left(1-\phi\left(x_1\right)\right)\left(1-\phi\left(x_2\right)\right)\right)\\ z&=x_1+x_2\end{aligned}\right\}\tag{2}$$

其中

$$\phi(x)=\begin{cases}1-\dfrac{1}{\sqrt{4\pi x}}\displaystyle\int_{-\infty}^{+\infty}\tanh\left(\frac{\tau}{2}\right)\mathrm{e}^{\left(\frac{-(\tau-x)^2}{4x}\right)}\mathrm{d}\tau, & x>0\\ 1, & x=0\end{cases}\tag{3}$$

高斯近似函数的近似推导和提出第1次出现在文献[18]，然后该函数被应用于极化码的高斯近似构造中。现在基于高斯近似构造的极化码算法都基本上沿用了该公式，该高斯近似函数如式(4)所示

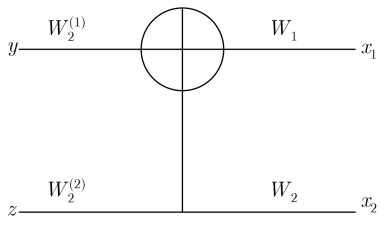


图 1 基础的蝶形计算结构

$$\phi(x)=\begin{cases}\exp\left(-0.4527x^{0.86}+0.0218\right), & 0<x<10\\ \sqrt{\dfrac{\pi}{x}}\exp\left(-\dfrac{x}{4}\right)\left(1-\dfrac{10}{7x}\right), & x\geq 10\end{cases} \tag{4}$$

3　基于改进高斯近似的极化码打孔算法

GAPPC算法主要思想是通过改进信道构造环节，提升打孔算法性能。首先引入高斯修正因子，改进高斯近似函数，得到子信道的可靠性排序集合。其次依据信道容量关系推导出改进的信道映射规则，对选出的无能力比特集合进行映射得到打孔比特集合，并结合可靠性排序集合完成打孔极化码构造，最后给出GAPPC具体算法流程。

3.1　改进高斯近似函数

极化码打孔算法的核心是去掉LLR值较低的比特位，因此需要在信道构造环节尽量降低高斯近似函数输出值。在经典高斯近似函数基础上，引入λ_0, λ_1, λ_2以及α作为修正因子，改进高斯近似函数为式(5)

$$\phi(x)=\begin{cases}\mathrm{e}^{\lambda_0 x^{\lambda_1}+\lambda_2}, & 0<x<10\\ \sqrt{\dfrac{\pi}{\alpha}}\mathrm{e}^{-\frac{x}{4}}\left(1-\dfrac{10}{7x}\right), & x\geq 10\end{cases} \tag{5}$$

$\phi(x),x\in(0,10)$用$\phi_l(x)$表示，$\phi(x),x\in[10,+\infty)$用$\phi_h(x)$表示，根据高斯函数的连续性和一致性，有　$\phi_l(x)|_{x\to 10^-}=\phi_h(x)|_{x\to 10^+}$　和$\phi_l{}'(x)\big|_{x\to 10^-}=\phi_h{}'(x)\big|_{x\to 10^+}$，同时改进的高斯近似函数值在$x=0$处应与GA函数保持一致，推导出高斯修正因子$\lambda_0,\lambda_1,\lambda_2$以及$\alpha$之间的关系满足式(6)

$$\left.\begin{aligned}\lambda_0&=\frac{1}{24\lambda_1}+\lambda_2\\ \lambda_1&=\frac{1}{24\left(\log_2\left(\sqrt{\dfrac{\pi}{\alpha}}\mathrm{e}^{-2.5\frac{6}{7}}\right)-\lambda_2\right)}\\ \lambda_2&=0.0218\end{aligned}\right\} \tag{6}$$

根据式(6)可知，λ_2为一个常数，因子λ_0,λ_1由α与λ_2表示，由此可知α是高斯修正因子的关键。由式(2)，改进高斯近似函数的输出值y如式(7)所示

$$y=\phi^{-1}\left(1-\left(1-\phi(x_1)\right)\left(1-\phi(x_2)\right)\right) \tag{7}$$

$\phi^{-1}(x)$函数是$\phi(x)$函数的反函数，因此两者单调性一致，$\phi(x)$对α的导函数为

$$\phi'(x)=\begin{cases}\left(\lambda'_0 x^{\lambda_1}+\lambda_0\lambda_1 x^{\lambda_1-1}\lambda_1{}'\right)\mathrm{e}^{\lambda_0 x^{\lambda_1}}, & 0<x<10\\ \left(-\dfrac{1}{2}\right)\sqrt{\pi}\alpha^{-\frac{3}{2}}\mathrm{e}^{-\frac{x}{4}}\left(1-\dfrac{10}{7x}\right), & x\geq 10\end{cases} \tag{8}$$

另外，式(6)中的λ_0,λ_1对α的求导结果为

$$\left.\begin{aligned}\lambda_0{}'&=\frac{-\lambda_1{}'}{24{\lambda_1}^2}\\ \lambda_1{}'&=\frac{1}{48\alpha\left(\log_2\left(\sqrt{\dfrac{\pi}{\alpha}}\mathrm{e}^{-2.5\frac{6}{7}}\right)-\lambda_2\right)^2}\end{aligned}\right\} \tag{9}$$

由式(9)，得出$\lambda_0{}'<0$，$\lambda_1{}'>0$，结合式(8)易知$\phi'(x)<0$，$\phi(x)$为单调递减函数，其反函数$\phi^{-1}(x)$也是单调递减函数。令$T=1-(1-\phi(x_1))(1-\phi(x_2))$，因为$\phi(x)\in(0,1)$，所以$T\in(0,1)$。式(7)中高斯近似函数的输出$y$对$\alpha$求导，得到式(10)

$$y'=\frac{1}{\phi(T)'}\left(\phi(x_1)'\left(1-\phi(x_2)\right)+\phi(x_2)'\left(1-\phi(x_1)\right)\right) \tag{10}$$

因为$\phi(x)'<0$，可得$\phi(x_1)'<0$，$\phi(x_2)'<0$和$\phi(T)'<0$，由式(10)，可以得出$y'>0$。

对于打孔算法来说，如果比特x_i被打孔，那么$L(y_i)$为0。在蝶形结构中，如果$L(y_i)$为0，可以得到$L(u_1)=0$和$L(u_2)=L(y_2)$(或者$L(u_2)=L(y_1)$)。如果两个信道都被打孔，那么会有$L(u_1)=L(u_2)=0$。在极化码编码结构中，从右往左进行蝶形计算的迭代会导致信道降级，这意味着相关信道会变差。由文献[17]可知，相较于未打孔版本，一旦某个比特被打孔，那么所有分离信道将会发生信道降级。因此在信道构造时候把打孔造成的信道降级考虑进去，降低$L(u_1)$，构造出降级的信道。在构造出的所有降级子信道中，受信道降级影响小的信道LLR降低得少，相对也更加稳定和可靠，反之亦然。考虑到打孔会使得打孔比特所在的信道LLR直接降低为0，因此改进的构造函数输出值y应该尽量降低，相应的因子α也应该降低。

由式(7)所知，如果y降低，$\phi(x)$的输出值应该增大，即$\phi_h(x)$增大：$\phi_h(x)>\sqrt{\dfrac{\pi}{x}}\mathrm{e}^{-\frac{x}{4}}\left(1-\dfrac{10}{7x}\right)$。为了保证不等式成立，满足$\phi(10)<\phi(0)$以及函数收敛快于牛顿迭代法，得可行性范围为$\alpha\in[1,10)$。取$\alpha$值为1，则根据式(5)可得到$\lambda_0=-0.16358$，$\lambda_1=1.1092$，因此改进的高斯近似函数如式(11)所示

$$\phi(x)=\begin{cases}\mathrm{e}^{-0.16358x^{1.1092}+0.0218}, & 0<x<10\\ \sqrt{\pi}\mathrm{e}^{-\frac{x}{4}}\left(1-\dfrac{10}{7x}\right), & x\geq 10\end{cases} \tag{11}$$

3.2　信道映射

由图1，用W_1,W_2表示x_1,x_2所在信道，经过蝶形计算后的信道表示为$W_2^{(1)}$和$W_2^{(2)}$。蝶形计算过程中信道容量关系[5]如式(12)所示

$$\left.\begin{aligned} &I\left(W_2^{(1)}\right) \le \min\left(I\left(W_1\right), I\left(W_2\right)\right) \\ &I\left(W_2^{(2)}\right) \ge \max\left(I\left(W_1\right), I\left(W_2\right)\right) \\ &I\left(W_2^{(1)}\right) + I\left(W_2^{(2)}\right) = I\left(W_1\right) + I\left(W_2\right) \end{aligned}\right\} \tag{12}$$

信道极化过程中信道容量始终满足$0 \le I(W) \le 1$，但如果x_i被打孔，则$I(W_i)$会降为0。根据上述式(12)，可以分析出信道极化过程中信道容量的映射关系，原始信道容量合并时候会有特定对应的信道容量，然而，分离的信道容量与合并前的信道容量并非一一对应。

如图2所示，对于蝶形计算中原始信道容量$I(W_i)$的各种情况，均有相应的分离信道容量与之对应，反之，却不成立。从分离信道容量$I\left(W_2^{(i)}\right)$角度出发，如果$I\left(W_2^{(1)}\right)=0, I\left(W_2^{(2)}\right)>0$，有两种不同的原始信道容量情况：$I(W_1)=0, I(W_2)>0$或者$I(W_1)>0, I(W_2)=0$，但如果$I\left(W_2^{(1)}\right)>0$, $I\left(W_2^{(2)}\right)=0$，则找不到原始信道容量的对应情况。因此，提出如下改进的信道映射规则：

(1) 如果$I\left(W_2^{(1)}\right)>0, I\left(W_2^{(2)}\right)>0$，可得到$I(W_1)>0, I(W_2)>0$；

(2) 如果$I\left(W_2^{(1)}\right)=0, I\left(W_2^{(2)}\right)>0$，可得到$I(W_1)=0, I(W_2)>0$；

(3) 如果$I\left(W_2^{(1)}\right)=0, I\left(W_2^{(2)}\right)=0$，可得到$I(W_1)=0, I(W_2)=0$；

(4) 如果$I\left(W_2^{(1)}\right)>0, I\left(W_2^{(2)}\right)=0$，不存在。

GAPPC算法在MGA构造之后使用改进的信道映射找到打孔比特位置，在x_1^N中选出无能力比特集合后，设其$I\left(W_N^{(i)}\right)=0$，然后应用所提的映射规则在极化码编码结构中进行迭代，找出最终打孔比特集合。

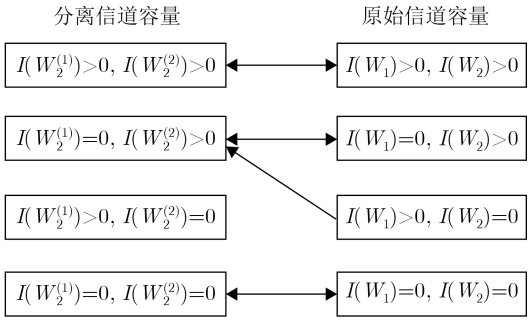


图 2 蝶形计算中信道容量关系

3.3 算法流程

将MGA引入构造，得到MGA构造，将MGA构造出的子信道按照升序排序得到的序列集合设为R_{MGA}。结合MGA构造和推导出的映射规则，提出GAPPC算法，算法具体如表1所示。

在GAPPC算法中，首先对N长的极化码进行MGA构造，按照LLR值升序得出子信道的可靠性序列R_{MGA}。选出无能力打孔位置集合$\mathcal{Q}=\{1,2,\cdots,P\}$，即$u_1^N$的前$P$比特作为无能力比特，$P=N-M$，应用$\mathcal{Q}$和改进的映射规则在极化码编码结构中迭代，找出打孔比特集合$\mathcal{P}$。根据R_{MGA}和$\mathcal{Q}$，获取无能力比特之外的最不可靠的$M-K$比特，作为剩余冻结集合$\mathcal{Q}^C$。根据$\mathcal{Q}^C$和$\mathcal{Q}$可以得到冻结比特集合A^C和信息比特集合A，完成对打孔极化码的构造。

在完成极化码构造之后，对N长码字u_1^N进行编码得到x_1^N，并运用打孔比特集合$\mathcal{P}$对编码后码字进行打孔得到M长传输码字y_1^M，得到(N,M,K)打孔极化码，码率为K/M。在传输过程中打孔比特不会被发送，在接收端，译码器会将打孔比特的LLR值设置为0并完成最终译码。

4 实验与结果分析

实验采用二进制相移键控调制信号源信号，传输信道采用AWGN信道，极化码译码采用串行抵消译码算法。实验中对比了LCP算法、WQP算法、文献[10]的打孔算法(BD算法)以及所提GAPPC算法之间的误码率(Bit Error Rate, BER)和误帧率(Frame Error Rate, FER)性能。实验中使用的极化码码长为512和256，使用的码率为2/3,1/2和1/4。另外，每次实验的最大模拟帧数为10^7，如果有1000个错误帧或共传输了10^7帧，实验将会停止。

图3展示了码率为1/2，码长分别为512和256下，4种极化码打孔算法的FER和BER性能对比。图3(a)显示了极化码码长为512，打孔后码长为372，码率为1/2时，GAPPC算法与LCP算法、WQP算法和BD算法的性能对比。相对于WQP算法、

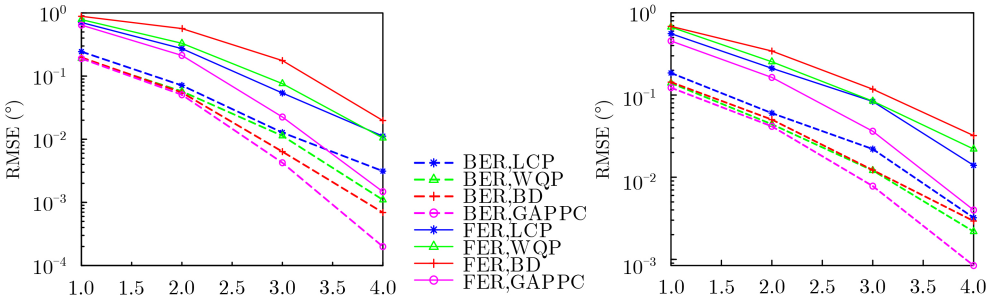
表 1 GAPPC算法

输入：极化码的码长N，打孔后码长M，信息比特长度K，打孔比特长度$P=N-M$
(1) 使用对N长的极化码进行MGA构造，获得子信道可靠性升序集合R_{MGA}
(2) 选出无能力比特位置集合$\mathcal{Q}=\{1,2,\cdots,P\}$，应用改进信道映射在编码结构中找出打孔比特位置集合$\mathcal{P}$
(3) 在R_{MGA}中除去无能力比特位置找出可靠性最低的$M-K$比特，设为剩余冻结比特集合$\mathcal{Q}^C$
(4) 依据$\mathcal{Q}^C$和$\mathcal{Q}$得出冻结比特集合A^C和信息比特集合A

LCP算法和BD算法，GAPPC算法的BER性能在SNR为1～4 dB时均有一定的提升，且BER在10^{-3}可以获得至少0.3 dB的性能增益。而GAPPC算法的FER性能明显优于其他3种打孔算法，FER在10^{-2}至少获得0.75 dB的性能增益。图3(b)显示了在极化码码长为256，打孔后码长为186，码率同样为1/2时，GAPPC算法的FER和BER性能均明显优于其他3种算法，BER在10^{-2}至少可以获0.25 dB的性能增益，FER在10^{-2}至少可以获得0.5 dB的性能增益。GAPPC算法在选择信息集合和冻结集合时使用的是基于MGA构造的子信道，依据子信道的LLR值获得可靠性排序集合，选择可靠性高的信道传输信息比特。这样构造出来的极化码能降低打孔带来的信道降级影响，从而可以获得更好的性能。由以上分析可知，在不同码长相同码率下GAPPC算法具有显著的性能提升，且极化码码长越大，算法性能提升越多。

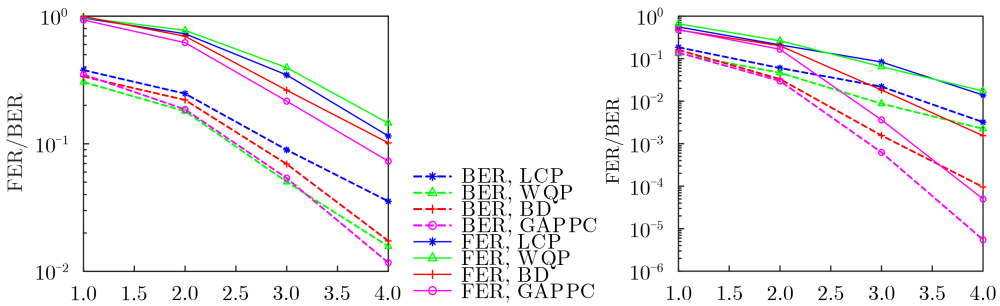
图4展示在码长为512，码率分别为2/3和1/4时，4种算法之间的FER和BER性能对比。图4(a)显示了在极化码码长为512，打孔后码长为360，码率为2/3时，GAPPC算法与LCP算法、WQP算法和BD算法的性能对比。在SNR为1～3 dB时，GAPPC算法与WQP算法的BER相近，明显优于LCP算法和BD算法；而在SNR为4 dB时，GAPPC算法的BER性能优于另外3种算法。而对于FER性能，GAPPC算法在SNR为1～4 dB时均具有显著的性能优势。图4(b)显示了在极化码码长为512，打孔后码长为400，码率为1/4时，GAPPC算法的BER与FER性能具有显著的提升。在SNR为4 dB时，GAPPC算法的FER可以达到5×10^{-5}，BER达到6×10^{-6}，相对于LCP算法、WQP算法和BD算法，GAPPC算法性能提升10倍以上。再结合图3(a)的实验结果可知，在相同码长不同码率下，GAPPC算法均可以获得显著性能增益，且码率越小，性能增益越大。

表2显示了不同打孔情况下，WQP算法的复杂度略高于LCP算法的复杂度，BD算法与LCP算法复杂度一致，而GAPPC算法的复杂度在4种算法中最高，其中LCP算法、WQP算法和BD算法都需要GA构造和比特倒置排序。由文献[16]可知，GA构造的复杂度为$O(N\lg N)$。LCP算法中，通过比特倒置排序得到的打孔比特集合会被预先设定，LCP算法只需要进行选择即可获得打孔比特集合，因此算法复杂度为$O(N\lg N)+O(1)$。而WQP算法选取S位无能力比特后进行比特倒置操作，因此算法复杂度为$O(N\lg N)+O(S)$, $S=N-M$。BD算法与LCP算法相似，同样只需对预设集合进行选择即可获取打孔比特集合，因此BD算法的复杂度与LCP



(a) (512, 372, 186)打孔极化码　　(b) (256, 186, 93)打孔极化码

图 3 不同码长相同码率下的4种极化码打孔算法性能对比



(a) (512, 360, 240) 打孔极化码　　(b) (512, 400, 100) 打孔极化码

图 4 相同码长不同码率下的4种极化码打孔算法性能对比

表 2 4种算法的计算复杂度对比

打孔算法	(256, 186, 93)	(512, 400, 100)	(1024, 860, 512)	(2048,1900,1024)
LCP	2049	4609	10241	22529
WQP	2118	4720	10404	22676
BD	2049	4609	10241	22529
GAPPC	2608	5616	11880	24156

算法一致。二者的区别在于BD算法的预设集合是利用二进制控制特性[10]排序后经过比特倒置操作获得的，而LCP算法的预设集合则是利用自然序与比特倒置操作获取的。预设集合在算法进行前完成设定，因此不计入算法复杂度。GAPPC算法的复杂度主要来自MGA构造以及信道映射，MGA构造的复杂度与GA构造的复杂度相同，都是$O(N\lg N)$，而信道映射的复杂度为$O(S\lg N)$，因此算法的复杂度为$O(N\lg N)+O(S\lg N)$。根据上述分析可知，相比于其他3种算法，GAPPC算法由于信道映射而具有更高的复杂性。

5 结束语

为了进一步提升极化码打孔算法的性能，在极化码打孔算法中联合考虑打孔和构造，提出高斯修正因子，并推导出最优的高斯修正因子和相应的MGA，依据信道容量的关系推导出改进的信道映射规则，找到打孔比特集合。基于MGA构造和改进的信道映射规则，设计了GAPPC算法并进行相关的实验验证。实验结果显示，在不同的码长和码率下，与LCP算法、WQP算法和BD算法相比，本文所提GAPPC算法的FER和BER性能均有显著提升，且码率越低码长越大获得的性能增益越多，但算法的复杂度会略有增加。

参 考 文 献

[1] ARIKAN E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels[J]. *IEEE Transactions on Information Theory*, 2009, 55(7): 3051–3073. doi: 10.1109/TIT.2009.2021379.

[2] 刘建航, 何怡静, 李世宝, 等. 基于预译码的极化码最大似然简化连续消除译码算法[J]. 电子与信息学报, 2019, 41(4): 959–966. doi: 10.11999/JEIT180324.
LIU Jianhang, HE Yijing, LI Shibao, *et al*. Pre-decoding based maximum-likelihood simplified successive-cancellation decoding of polar codes[J]. *Journal of Electronics & Information Technology*, 2019, 41(4): 959–966. doi: 10.11999/JEIT180324.

[3] 王琼, 罗亚洁, 李思舫. 基于分段循环冗余校验的极化码自适应连续取消列表译码算法[J]. 电子与信息学报, 2019, 41(7): 1572–1578. doi: 10.11999/JEIT180716.
WANG Qiong, LUO Yajie, and LI Sifang. Polar adaptive successive cancellation list decoding based on segmentation cyclic redundancy check[J]. *Journal of Electronics & Information Technology*, 2019, 41(7): 1572–1578. doi: 10.11999/JEIT180716.

[4] ESLAMI A and PISHRO-NIK H. A practical approach to polar codes[C]. 2011 IEEE International Symposium on Information Theory Proceedings (ISIT), St. Petersburg, Russia, 2011: 16–20. doi: 10.1109/ISIT.2011.6033837.

[5] SHIN D M, LIM S C, and YANG K. Design of length-compatible polar codes based on the reduction of polarizing matrices[J]. *IEEE Transactions on Communications*, 2013, 61(7): 2593–2599. doi: 10.1109/TCOMM.2013.052013.120543.

[6] NIU Kai, CHEN Kai, and LIN Jiaru. Beyond turbo codes: Rate-compatible punctured polar codes[C]. IEEE International Conference on Communications (ICC), Budapest, Hungary, 2013: 3423–3427. doi: 10.1109/ICC.2013.6655078.

[7] NIU Kai, DAI Jincheng, CHEN Kai, *et al*. Rate-compatible punctured polar codes: Optimal construction based on polar spectra[EB/OL]. https://arxiv.org/pdf/1612.01352, 2016.

[8] LIU Wei, WANG Yue, LI Ao, *et al*. An improved puncturing scheme for polar codes[C]. 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020: 154–158. doi: 10.1109/IWCMC48107.2020.9148522.

[9] HANIF M A and VAFI S. A modified approach to punctured product polar codes[J]. *Journal of Telecommunications and Information Technology*, 2019, 3: 63–69. doi: 10.26636/jtit.2019.132219.

[10] JANG M, AHN S K, JEONG H, *et al*. Rate matching for polar codes based on binary domination[J]. *IEEE Transactions on Communications*, 2019, 67(10): 6668–6681. doi: 10.1109/TCOMM.2019.2930502.

[11] HONG S N and JEONG M O. An efficient construction of rate-compatible punctured polar (RCPP) codes using hierarchical puncturing[J]. *IEEE Transactions on Communications*, 2018, 66(11): 5041–5052. doi: 10.1109/TCOMM.2018.2854183.

[12] ZHAO Jianhan, ZHANG Wei, LIU Yanyan, *et al*. A rate-matching concatenation scheme of polar codes with outer

reed-Solomon codes[J]. *IEEE Wireless Communications Letters*, 2021, 10(3): 459–463. doi: 10.1109/LWC.2020.3033850.

[13] ZHANG Liang, ZHANG Zhaoyang, WANG Xianbin, *et al*. On the puncturing patterns for punctured polar codes[C]. 2014 IEEE International Symposium on Information Theory (ISIT), Honolulu, USA, 2014: 121–125. doi: 10.1109/ISIT.2014.6874807.

[14] BIOGLIO V, GABRY F, and LAND I. Low-complexity puncturing and shortening of polar codes[C]. 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, USA, 2017: 1–6. doi: 10.1109/WCNCW.2017.7919040.

[15] LI Liping, SONG Wei, and NIU Kai. Optimal puncturing of polar codes with a fixed information set[J]. *IEEE Access*, 2019, 7: 65965–65972. doi: 10.1109/ACCESS.2019.2918346.

[16] WU Daolong, LI Ying, and SUN Yue. Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation[J]. *IEEE Communications Letters*, 2014, 18(7): 1099–1102. doi: 10.1109/LCOMM.2014.2325811.

[17] TAL I and VARDY A. How to construct polar codes[J]. *IEEE Transactions on Information Theory*, 2013, 59(10): 6562–6582. doi: 10.1109/TIT.2013.2272694.

[18] CHUNG S Y, RICHARDSON T J, and URBANKE R L. Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation[J]. *IEEE Transactions on Information Theory*, 2001, 47(2): 657–670. doi: 10.1109/18.910580.

李世宝：男，1978年生，硕士、副教授，研究方向为移动计算、信道编码等.

高　迅：男，1996年生，硕士生，研究方向为信道编码.

董振威：男，1997年生，硕士生，研究方向为信道编码.

刘建航：男，1978年生，博士、副教授，研究方向为车联网等.

崔学荣：男，1979年生，博士、教授，研究方向为智能感知等.

责任编辑：余　蓉