光电技术及应用

DOI：10.16818/j.issn1001－5868.2022100501

一种基于错误集的极化码改进 SCL 译码算法

袁建国[1*]，张降龙[2]，张帅康[1]，王煦杰[1]，莫珍珠[1]

（1. 重庆邮电大学 光电工程学院，重庆 400065；2. 重庆邮电大学 通信与信息工程学院，重庆 400065）

摘　要： 针对极化码在中短码长时纠错性能的不足，提出了一种基于错误集的极化码改进串行抵消列表（Successive Cancellation List of Polar Codes Based on Error Set，ES-SCL）译码算法。该算法首先根据极化码的信道特性构造错误集，在极化码编码时根据错误集中的元素设置奇偶校验（Parity Check，PC）位，其余位置则放置信息比特和冻结比特，译码器在译码 PC 位时，每条路径通过校验函数得到 PC 位的比特估计，不执行路径分裂和剪枝，其余位置则执行 SCL 译码。仿真结果表明，在加性高斯白噪声信道下，当码长为 512，码率为 0.5，误块率为 10^{-5}，最大译码列表数为 8 时，相较于 PC-PSCL 译码算法以及 CA-SCL 译码算法，所提出的 ES-SCL 译码算法获得了约 0.18 和 0.15 dB 的增益；当码长为 256，码率为 0.5，误码率为 10^{-5}，最大译码列表数为 8 时，相较于 CA-SCL，PC-PSCL 译码算法，获得了约 0.3 和 0.35 dB 的增益；此外，采用部分比特分裂译码的 ES-SCL 译码算法可以在误块率与 PC-PSCL 译码算法几乎相同的情况下，减少约 50％的排序次数，具有更低的译码复杂度。

关键词： 极化码；奇偶校验码；错误集；SCL 译码

中图分类号： TN919　**文章编号：** 1001－5868(2023)01－0109－06

An Improved SCL Decoding Algorithm of Polar Codes Based on Error Set

YUAN Jianguo[1]，ZHANG Xianglong[2]，ZHANG Shuaikang[1]，WANG Xujie[1]，MO Zhenzhu[1]

（1. School of Optoelectronic Engineering，Chongqing University of Posts and Telecommunications，Chongqing 400065，CHN；
2. School of Commun. and Information Engin.，Chongqing University of Posts and Telecommunications，Chongqing 400065，CHN）

Abstract： In order to improve the error-correction performance of polar codes with the medium and short code length，an improved decoding algorithm of successive cancellation list of polar codes based on error set (ES-SCL) is proposed. The algorithm first constructed an error set according to the channel characteristics of polar codes. Then the parity check (PC) bits were set according to the elements in the error set during the encoding of polar codes and the information bits and frozen bits were placed in the remaining positions. When the decoder decoded the PC bits，each path did not perform the path splitting and pruning，but could get the estimation of the PC bits through the check function. For the other bits，the decoder would directly perform the SCL decoding. Simulation results show that under the additive white Gaussian noise channel，when the code length is 512，the code rate is 0.5，the block error rate is 10^{-5} and the maximum list length is 8，compared with the PC-PSCL and CA-SCL decoding algorithm，the proposed ES-SCL decoding algorithm can achieve a gain of about 0.18 and 0.15 dB. When the code length is 256，the code rate is 0.5，the block error rate is 10^{-5} and the maximum list length is 8，compared with the CA-SCL and PC-PSCL decoding algorithm，the proposed ES-SCL decoding

收稿日期：2022－10－05.

基金项目：国家自然科学基金项目(61971079)；重庆市研究生科研创新项目(CYS21301)；2021 年大学生科研训练计划(A2021-68).

* **通信作者：**袁建国　**E-mail：** yyyyjg@126.com

algorithm can achieve a gain of about 0.3 and 0.35 dB. Furthermore, the ES-SCL decoding algorithm that uses the partial bit splitting decoding can reduce the sorting number of about 50% when the block error rate is almost the same as that of the PC-PSCL decoding algorithm, thus its decoding complexity is lower.

Key words: polar codes; parity check codes; error sets; SCL decoding

0 引言

极化码由 Arikan 提出,是第一类能够被证明达到二进制输入无记忆对称信道(Symmetric Binary-input Memoryless Channels)的信道容量的码字,是新一代移动通信系统的重要技术。极化码已经成功入选 5G 标准,成为 5G 增强移动宽带场景(Enhanced Mobile BroadBand)下控制信道的编码方案,是当前信道编码领域的研究热点[1]。然而在码长有限的情况下,由于信道极化不完全,实际 SC 译码算法性能并不理想[1-3]。于是文献[2]提出了能够保留多条译码路径的串行抵消列表(Successive Cancellation List,SCL)译码算法,文献[3]采用与循环冗余校验(Cyclic Redundancy Check,CRC)码级联的 CRC-Polar 码,提出循环冗余校验码辅助串行抵消列表(CRC Aid SCL,CA-SCL)译码算法,提高了 SCL 译码算法的译码性能。但由于 CRC 码一般添加在信息序列的末尾,无法对译码过程中出现的错误进行检测,因此有学者提出级联奇偶校验(Parity Check,PC)码的方案[4-5]来及时地纠正译码过程中出现的错误,进一步提高极化码的纠错能力。奇偶校验码级联极化码方案的关键问题在于 PC 码位置的选取以及确定 PC 位的校验函数。文献[4-5]均通过高斯近似[6]得到每个信道的错误概率 P_e 来确定 PC 码的放置位置,不同的是文献[4]中的奇偶校验码辅助串行抵消列表(Parity Check Aided Successive Cancellation List,PC-SCL)译码算法选择 P_e 最大的非冻结位作为 PC 码的位置,文献[5]中的基于奇偶校验码级联极化码的串行抵消局部列表(Parity Check Aided Partial Successive Cancellation List,PC-PSCL)译码算法选择在部分分裂比特中 P_e 最小的位置作为 PC 码的位置。这是由于文献[4]中的 PC-SCL 译码算法在译码 PC 位置时,每条译码路径不再进行路径分裂而是仅根据校验函数计算出每个 PC 位的比特估计值,此时 PC 码的效果类似于冻结位,因此需要将 PC 码放在最不可靠的位置上。而文献[5]的 PC-PSCL 译码算法在译码 PC 位置时,每条路径依然采用极化码的 SC 译码得到 PC 位的比特估计值,再验证 PC 位的估计值是否通过 PC 校验函数,最后仅保留通过校验的路径。当正确路径上的 PC 码出现译码错误时,正确路径会因为不满足校验函数而被删除掉,为了减少这种情况的发生,于是文献[5]需要将 PC 码放在最可靠的位置上。由于文献[4-5]仅依靠高斯近似选取 PC 码的位置,未充分考虑极化码的信道特性,因此相较于 CA-SCL 译码算法未取得明显的性能提升。针对该问题,本文通过分析极化码中极化信道的特性构造出错误集(Error Set,ES),并根据错误集中的元素设置 PC 位,进而提出一种误块率更低的基于错误集的极化码串行抵消列表(Successive Cancellation List of Polar Code Based on Error Set,ES-SCL)译码算法。

1 极化码译码

1.1 极化码 SC 译码

极化码的 SC 译码算法是序贯译码算法,从首个比特开始逐个译码信源比特,直到译完最后一个比特为止。

设 W 是二进制输入无记忆对称信道,输入符号集合为 $X\in\{0,1\}$,输出符号集合为 Y,信道转移概率记为 $W(y|x)$,其中 $x\in X=\{0,1\}$,$y\in Y$。信道 W 复用 N 次,以传输码字序列 $x_1^N=u_1^N\boldsymbol{G}_N$,其中 u_1^N 表示输入至极化码编码器的信源序列,$\boldsymbol{G}_N$ 表示极化码生成矩阵。接收序列记为 y_1^N,使用对数似然比(Logarithm Likelihood Ratio,LLR)作为译码过程中的可靠性度量。极化信道 $W_N^{(i)}(y_1^N,u_1^{i-1}|u_i)$ 的 LLR 定义如下:

$$L_N^{(i)}(y_1^N,u_1^{i-1}\mid u_i)=\ln\frac{w_N^i(y_1^N,u_1^{i-1}\mid u_i=0)}{w_N^i(y_1^N,u_1^{i-1}\mid u_i=1)} \tag{1}$$

使用 LLR 作为译码度量时,比特判决准则如下:

$$u_i=\begin{cases}0, i\in A^c \text{ 或 } LLR>0\\ 1, i\in A \text{ 或 } LLR<0\end{cases} \tag{2}$$

其中,A^c 表示极化码冻结比特集合,通常将其值置

为 0，A 表示极化码非冻结比特集合。

1.2　极化码 CA-SCL 译码

CA-SCL 译码器在译码每一个非冻结比特位置时进行路径分裂，分别保存译码可能结果 0 和 1，并按式(3)所示更新每条路径的路径度量值(Path Metric，PM)[7]。

$$PM_l^i = PM_l^{i-1} + \ln(1+\mathrm{e}^{-(1-2\hat{u})_{i,l}})L_N^i(y^N, u_{1,l}^{i-1}) \tag{3}$$

在实际应用中，可使用式(4)来简化表示：

$$PM_l^i = \begin{cases} PM_l^{i-1}, \hat{u}_{i,l} = \dfrac{1}{2}(1-\mathrm{sgn}(L_N^i(l))) \\ PM_l^{i-1} + |L_N^i(l)|, \text{其他} \end{cases} \tag{4}$$

其中，PM_l^i 表示第 l 条译码路径在译码第 i 个比特后的路径度量值，$\hat{u}_{i,l}$ 表示第 l 条译码路径中第 i 个比特的译码估计值。当分裂后的路径数超过最大译码列表数 L 时，仅保留 L 条 PM 较小的路径。在译完最后一个比特后，从通过 CRC 校验的路径中选择 PM 最小的路径作为 CA-SCL 译码器的输出码字。

2　CA-SCL 译码失败原因分析

CA-SCL 译码失败的原因有两种：译码过程中丢失掉正确译码路径以及在译码结束后正确译码结果未被选出来。对于后一种情况可以通过优化 CRC 码的生成多项式[8]以及增加 CRC 码的位数得到有效的改善，本文主要对前一种情况进行分析。

假设发送信息序列为全 0 码字，当前译码第 K 个非冻结比特，在译完第 $K-1$ 个非冻结比特时，存活的路径数量为 $m(m \leqslant L)$，并且第 m 条路径中保留着正确译码结果，左侧为每条路径在译码第 K 个非冻结比特前的 PM，右侧为路径分裂后的 PM，如图 1 所示。

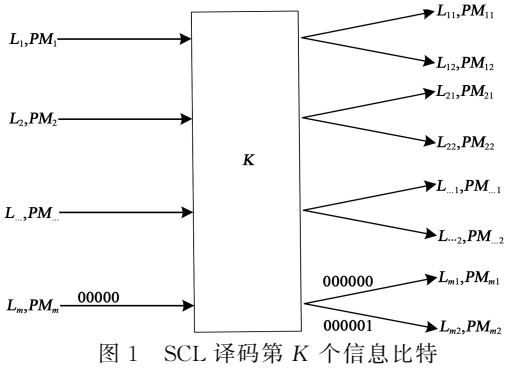


图 1　SCL 译码第 K 个信息比特

设路径 L_m 为译码第 K 个非冻结比特前的正确路径，由 SCL 的译码过程可知，如果在译完第 K 个非冻结比特后出现了正确路径丢失，那么有以下两种可能情况。

情况 1：路径 L_{m1} 没有保留下来但是路径 L_{m2} 保留了下来。此时一定有 $PM_{m1} > PM_{m2}$，在图 1 中表现为第 m 条译码路径在第 K 个比特有 $L_N^k < 0$，即出现了和正确译码结果相反的 LLR。

情况 2：路径 L_{m1} 与 L_{m2} 都没有保留下来。此时 PM_{m1} 与 PM_{m2} 的大小关系不确定。

通过仿真实验统计得到情况 1 出现的概率如图 2 所示。可以看到，除了码长为 1 024 以外，情况 1 出现的概率都超过了 0.5，这表明 SCL 译码在大多数情况下是由于情况 1 导致正确路径丢失。而情况 1 只会发生在硬判决出错的位置，因此通过搜寻最容易出现硬判决错误的非冻结比特位置作为错误集(Error Set，ES)中的元素，然后将其中前 K 个元素设置为 PC 位，从而改善这些元素的译码成功率。

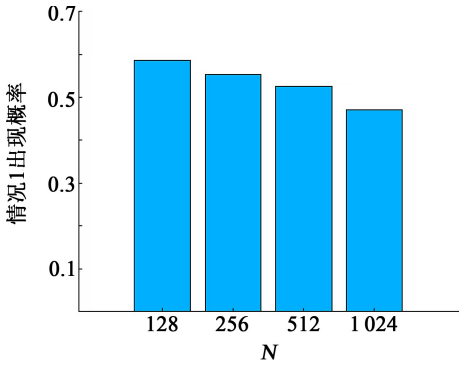


图 2　情况 1 出现的概率

3　错误集的构造

在构造错误集之前，需要介绍两个定理，设极化码码长为 N，其中 W_N^i 表示第 i 个极化信道的可靠度，有以下定理成立。

定理 1：极化码码长越长，W_N^1 越小，即第一个极化信道越不可靠。

证明：如图 3 所示，根据极化码的极化特性，两个信道 W 极化会得到一个更好的极化信道 W_2^2 和另一个更差的极化信道 W_2^1，而第一个信道始终是极化后更差的极化信道，因此码长越长，极化的次数越多，第一个极化信道越不可靠。

定理 2[9]：设极化码码长 $N \geqslant 4$ 则

$$W_N^1 < W_N^2 < W_N^3, \text{且 } W_N^i < W_N^j, \quad \forall i \in \{1,2,3\}, \forall j \geqslant 4 \tag{5}$$

定理 2 说明了在长度为 N 的极化码中，前三个极化信道 W_N^1, W_N^2, W_N^3 最不可靠。

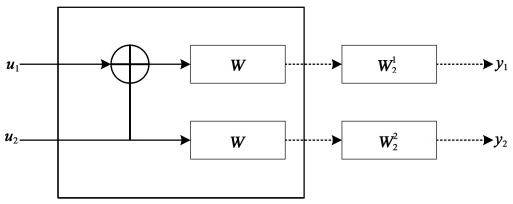


图 3 两个信道极化过程示意图

如图 4 所示，极化码可以看作由多个子极化码块组成，图中白色叶子节点代表冻结比特，黑色叶子节点代表信息比特，灰色节点代表既包含冻结比特又包含信息比特的子块，其中仅包含信息比特的子块{A，B，C，D，E}被称为码率 1（Rate-1，R1）节点[10]，文献[11]将所有 R1 节点的第一个比特 u_4，u_8，u_{10}，u_{11}，u_{13}，构成关键集（Critical Set，CS），并通过仿真实验验证了这种方式构造的关键集包含了约 99%的发生硬判决错误的比特。尽管关键集包含了绝大部分的不可靠比特，但关键集内元素的可靠度大小是未知的，而在选取 PC 位时往往需要对这些元素进行可靠性排序再从中选择出前 K 个最不可靠的比特。因此，本文通过对关键集进行改进提出一种错误集的构造方法。

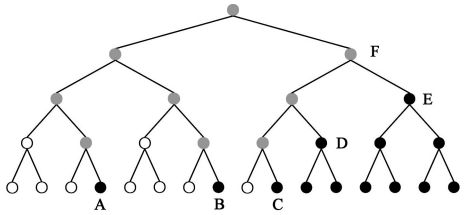


图 4 N 为 16 的极化码二叉树示意图

首先观察图 4 可知，对于部分 R1 节点{C，D，E}可以组合成为一个单奇偶校验（Single-Parity-Check，SPC）[12]节点 F。由定理 2 可知，此时 F 节点最不可靠的比特为 u_{10}，利用 R1 节点的合并，可以缩小一部分关键集中的元素。

接着由定理 1 可知，极化码码长越长，W_N^1 的可靠性越低，基于此，将 R1 节点和 SPC 节点通过节点长度进行降序排列，当出现两节点长度相同时，信道序号低的节点排在前面，最后依次将排序后的每个节点的第一个非冻结比特添加至错误集中。注意到此时每个 R1 节点和 SPC 节点的信道 W 虽然是不一样的，但是在构造错误集时可以近似的认为它们相等。错误集具体构造步骤如表 1 所示。

分析表 1 可知，错误集的构造首先需要遍历冻结比特集合 A^c 与非冻结比特集合 A，此步骤的时间复杂度为 $O(N)$，然后将 R1 节点和 SPC 节点按照定理 1 排序，假设此时待排序的节点总数为 t，那么此步骤的平均时间复杂度为 $O(t\log_2 t)$，则构造错误集的平均时间复杂度为 $O(N+t\log_2 t)$。

表 1 错误集的构造方法

错误集的构造方法
输入：A^c，A
输出：错误集 E
1：通过 A^c，A 确定 R1 节点和 SPC 节点；
2：将 R1 和 SPC 节点按节点长度降序排列，当出现两节点长度相等时，将信道序号低的节点排在前面；
3：将排序后的 R1 节点和 SPC 节点的第一个非冻结比特依次添加至错误集 E 中；

4 ES-SCL 译码算法

4.1 PC 位置的选取

设需要放置 PC 码的数量为 K，选择本文所构造的错误集中前 K 个元素作为 PC 码的位置，其中每个 PC 位的值 u_{pi} 由式(6)所示的校验函数计算所得。

$$u_{pi}=\left(\sum_{n\in S_i}u_n\right)\bmod 2 \tag{6}$$

其中，S_i 表示第 $i-1$ 个 PC 位至第 i 个 PC 位中间的信息比特集合。

4.2 编译码过程

设极化码码长为 N，传输的信息比特序列为 u_1^m，添加的 PC 码的数目为 k，添加的 CRC 码的数目为 r。

在编码时首先根据 4.1 选择 k 个 PC 码的位置以及计算每个 PC 位的值，完成 PC 码编码，得到此时编码后的序列 u_1^{m+k} 接着对 u_1^{m+k} 进行 CRC 编码得到 u_1^{m+k+r}，最后将 u_1^{m+k+r} 作为极化码中非冻结比特集合进行极化码编码得到码字 x_1^N。具体编码步骤如表 2 所示。

表 2 编码步骤

编码步骤
输入：N，u_1^m，k，r，错误集 E
输出：极化码码字 x_1^N
1：选取错误集 E 中前 k 个元素作为 PC 位；
2：对序列 u_1^m 进行 PC 编码，得到序列 u_1^{m+k}；
3：对序列 u_1^{m+k} 进行 CRC 编码，得到序列 u_1^{m+k+r}；
4：将序列 u_1^{m+k+r} 作为极化码的非冻结比特集合进行极化码编码，得到极化码码字 x_1^N；

在译码时，当译码器译码至 PC 位时，每条译码路径根据各自的译码结果通过校验函数计算得到 PC 位的译码估计值，并根据此时的 LLR 更新 PM。当译码器译码其他位置时，执行 SCL 译码，并在译

码结束后从通过 CRC 校验的路径中选择 PM 最小的路径作为译码器的输出结果。其具体步骤如表 3 所示。

表 3　译码步骤

译码步骤
输入：N，A^c，接收信号 y，PC 码位置集合 S
输出：信息序列估计 $\hat{u}_1^N$
1：FOR $i=1:N$
2：IF $i\in A^c$
3：每条路径上$\hat{u}_i=0$；
4：ELSE
5：IF $i\in S$
6：每条路径根据各自译码结果计算$\hat{u}_i$，并更新 PM；
7：ELSE
8：每条路径执行 SCL 译码得到$\hat{u}_i$；END IF；
9：END IF；
10：END FOR；
11：IF L 条路径中存在通过 CRC 校验的路径
12：选择其中 PM 值最小的路径作为译码的输出序列$\hat{u}_1^N$；
13：ELSE
14：宣布本次译码失败
15：END IF；

注意到在译码时 PC 位置的值是通过前面的译码结果计算得到，并没有执行路径分裂，此时 PC 码具有类似于冻结比特的作用，这样做的好处是可以消除情况 1 或情况 2 在该位置的发生，从而避免该位置出现译码失败。但此时不能简单地使用冻结比特来代替 PC 码，这是因为 PC 码相较于冻结比特会增大每条路径的码字差异，提高极化码的纠错性能，如图 5 所示。ES-SCL 译码算法整体编译码流程图如图 6 所示。

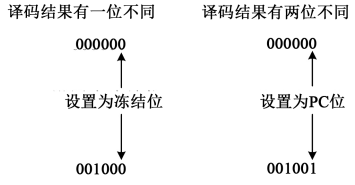


图 5　PC 码与冻结比特的比较

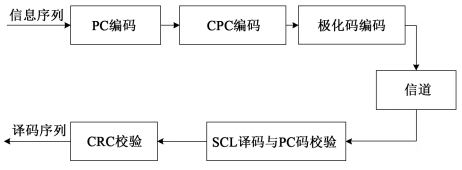


图 6　ES-SCL 编译码流程图

5　仿真分析

在本节中通过仿真实验分析不同算法的误块率以及各自的译码复杂度。由于文献[7]中 PC-PSCL 译码算法与文献[13]中 PSS-SS-SCL 译码算法在译码时仅对部分非冻结比特进行路径分裂，因而在本文所提出的 ES-SCL 译码算法基础上采用文献[11]的关键集进行部分比特分裂译码，并将部分分裂译码的 ES-SCL(本文简称为 PS-ES-SCL)译码算法与文献[5]中 PC-PSCL 译码算法和文献[13]中 PSS-SS-SCL 译码算法进行对比分析。分析 SCL 的译码过程可知，当译码列表总数大于 L 时，需要对 PM 进行排序，保留 L 条 PM 较小的路径。由于路径数量是呈指数式地增长，在译码第$(1+\log_2 L)$个非冻结比特时就需要进行排序并且排序工作几乎是每个比特译码时复杂度最高的部分，因此本文通过统计各算法平均一次译码的排序次数来描述各算法译码复杂度的高低。图 7 为码长 $N=512$，码率 $R=\frac{m}{N}=0.5$(m 为传输的信息比特数量)时各算法的误块率仿真结果图，图 8 为码长 $N=256$，码率 $R=\frac{m}{N}=0.5$ 时各算法的误块率仿真结果图。其他仿真参数均设置相同，即最大译码列表数 $L=8$，CA-SCL，PSS-SS-SCL 译码算法采用 16 位 CRC 校验码；PC-PSCL，ES-SCL 译码算法以及部分分裂的 ES-SCL 译码算法均采用 8 位 PC 码和 8 位 CRC 校验码。各算法均采用 BPSK 调制，信道采用加性高斯白噪声信道。图 9 统计了各算法在不同码长下平均一次译码的排序次数。

由图 7 和图 8 可知，本文所提出的 ES-SCL 译码算法的误块率均低于其他译码算法，在 $BLER=10^{-5}$，$N=512$ 时，ES-SCL 译码算法较 CA-SCL 译码算法约有 0.15 dB 的增益，在 $BLER=10^{-5}$，$N=256$ 时，ES-SCL 译码算法较 CA-SCL 译码算法约有 0.3 dB 的增益。并且从图 9 可以看出本文提出的 ES-SCL 译码算法的译码复杂度与 CA-SCL 译码算法的译码复杂度几乎相同，由于计算 PC 位的值仅涉及到模二加法运算，对总的译码复杂度影响不大，结合第 4 节的分析可知，ES-SCL 译码算法相较于 CA-SCL 译码算法仅增加了一次平均时间复杂度为 $O(N+t\log_2 t)$的错误集构造步骤而已。从图 7～9 中还可以看出，PS-ES-SCL 译码算法的误块率与文献[5]中的 PC-PSCL 译码算法的误块率几乎一致，但所需的排序次数降低了 50%，具有更低的译码复杂度。并且 PS-ES-SCL 译码算法相较于文献[13]中的 PSS-SS-SCL 译码算法在排序次数差不多的情况下拥有更低的误块率，进一步体现了本文算法在

提高极化码纠错性能上的优势。

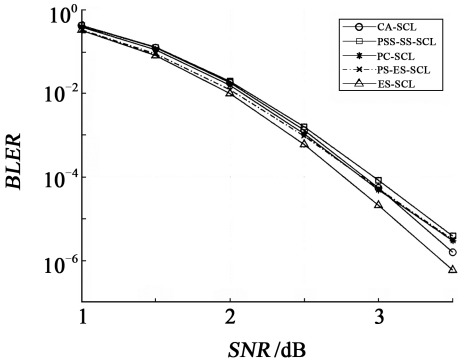


图 7 $N=512, R=0.5$ 时不同算法误块率仿真图

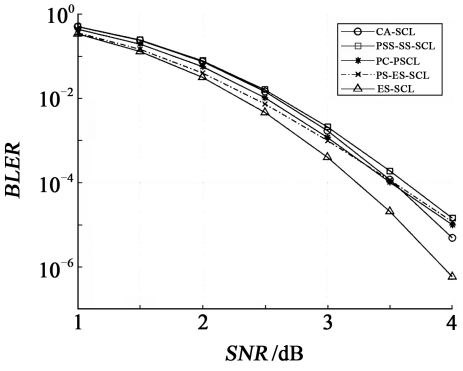


图 8 $N=256, R=0.5$ 时不同算法误块率仿真图

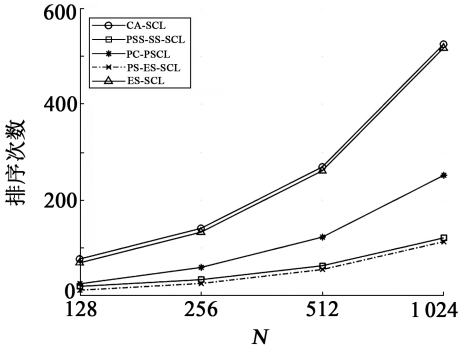


图 9 不同算法的平均排序次数

综上所述，本文所提出的 ES-SCL 译码算法在几乎不增加额外复杂度的情况下，相较于 CA-SCL，PC-PSCL 以及 PSS-SS-SCL 译码算法拥有更低的误块率，有效地提升了极化码的纠错性能。

6 结束语

本文针对极化码在中短码长时纠错性能的不足，提出一种基于错误集的奇偶校验码级联极化码算法，通过构造错误集来确定 PC 码的位置，利用 PC 码改善 SCL 译码过程中出现正确路径丢失的情况，并且增大了译码路径之间的码字差异。仿真结果表明：本文所提出的 ES-SCL 译码算法在几乎不增加额外复杂度的情况下，能有效地改善极化码的纠错性能。

参考文献：

[1] 刘 伟，段红光. 极化码自适应连续消除列表比特翻转译码算法[J]. 重庆邮电大学学报(自然科学版)，2021，33(1)：87-93.
Liu Wei, Duan Hongguang. Adaptive successive cancellation list bit-flip decoding of polar codes[J]. J. of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2021, 33(1): 87-93.

[2] Tal I, Vardy A. List decoding of polar codes[J]. IEEE Trans. on Information Theory, 2015, 61(5): 2213-2226.

[3] Cheng Fengyi, Liu Aijun, Zhang Yingxian, et al. CRC location design for polar codes[J]. IEEE Commun. Lett., 2018, 22(11): 2202-2205.

[4] Wang Tao, Qu Daiming, Jiang Tao. Parity-check-concatenated polar codes[J]. IEEE Commun. Lett., 2016, 20(12): 2342-2345.

[5] 刘顺兰，王 燕. 一种基于奇偶校验码级联极化码的低复杂度译码算法[J]. 电子与信息学报，2022，44(2)：637-645.
Liu Shunlan, Wang Yan. A low-complexity decoding algorithm based on parity-check-concatenated polar codes[J]. J. of Electronics & Information Technol., 2022, 44(2): 637-645.

[6] Trifonov P. Efficient design and decoding of polar codes[J]. IEEE Trans. on Communications, 2012, 60(11): 3221-3227.

[7] Balatsoukas-Stimming A, Parizi M B, Burg A. LLR-based successive cancellation list decoding of polar codes[J]. IEEE Trans. on Signal Processing, 2015, 63(19): 5165-5179.

[8] Zhang Qingshuang, Liu Aijun, Pan Xiaofei, et al. CRC code design for list decoding of polar codes[J]. IEEE Commun. Lett., 2017, 21(6): 1229-1232.

[9] 于永润. 极化码编译码技术研究[D]. 南京：东南大学，2019.
Yu Yongrun. Investigations on the construction and decoding of polar codes[D]. Nanjing: Southeast University, 2019.

[10] Alamdar-Yazdi A, Kschischang F R. A simplified successive-cancellation decoder for polar codes[J]. IEEE Commun. Lett., 2011, 15(12): 1378-1380.

[11] Zhang Z Y, Qin K J, Zhang L, et al. Progressive bit-flipping decoding of polar codes over layered critical sets[C]// 2017 IEEE Global Commun. Conf., 2017: 1-6.

[12] Hanif M, Ardakani M. Fast successive-cancellation decoding of polar codes: Identification and decoding of new nodes[J]. IEEE Commun. Lett., 2017, 21(11): 2360-2363.

[13] Gao C Y, Liu R K, Dai B, et al. Path splitting selecting strategy-aided successive cancellation list algorithm for polar codes[J]. IEEE Commun. Lett., 2019, 23(3): 422-425.

作者简介：

袁建国(1968—)，男，博士，教授，研究方向为通信系统中极化码与 LDPC 码的编译码技术。