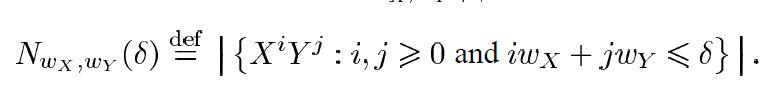
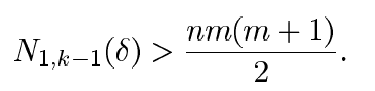
**ASD算法**

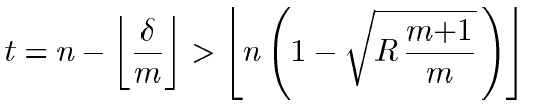
1. KV原版论文
2. 不小于特定weighted-degree项数的定义：



1. 插值时，参与插值的多项式项数与重数间的约束关系（如此线性方程组才有解）：



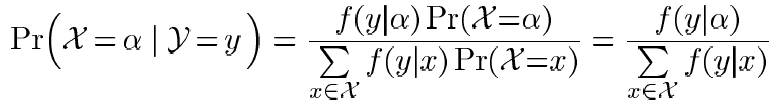
1. 纠错半径与重数、多项式weighted-degree的关系：

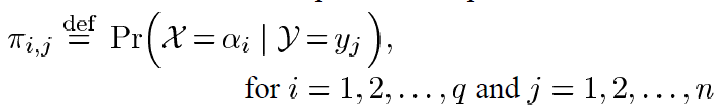


1. 概率测度：

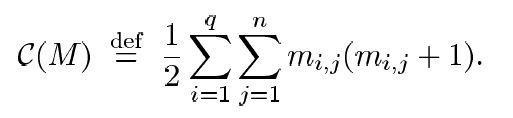
1639450311(1)

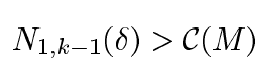
1. 贝叶斯定理下的后验概率：





1. 参与插值的多项式项数与重数间的约束关系可以重新写为：

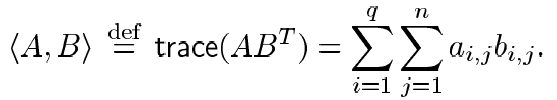




且新定义函数：

1639450610(1)

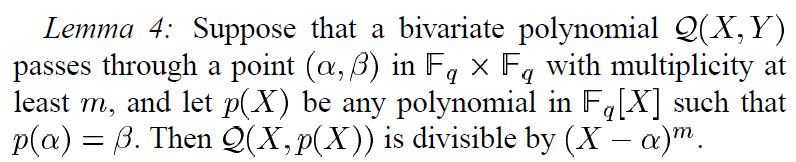
1. 定义内积测度：



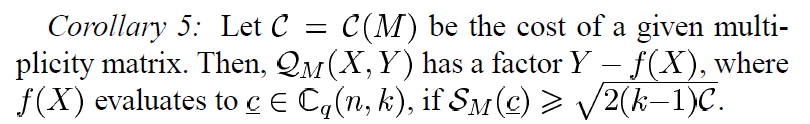
1639450676(1)

附带说一下，内积测度与互相关计算有关联。

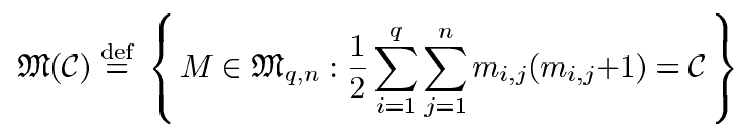
1. 说明重数插值的现象：



以及插值问题与重数及其约束的关系：



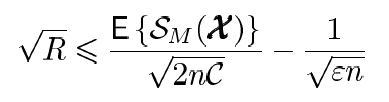
1. 定义M矩阵：



译码中的重数分配问题本质上就是最大化译码成功概率：

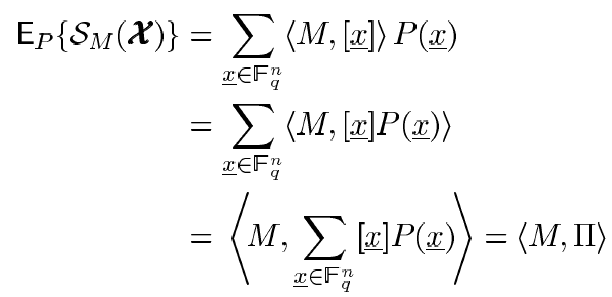
1639450961(1)

1. 对于任意小的错误概率，有：

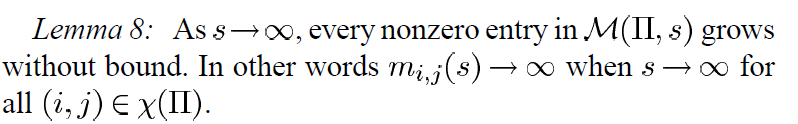


因此码长n趋于无穷时，本问题即要最大化分母上的数学期望

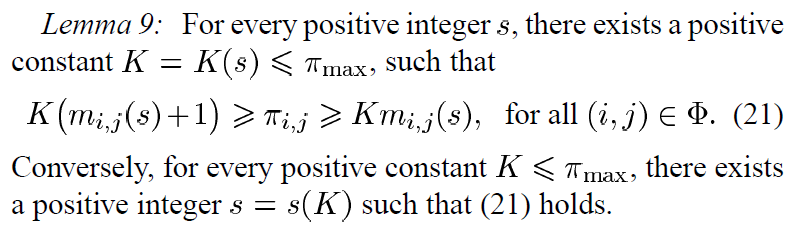
1. 此数学期望即：

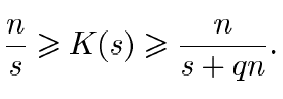


1. 文中宣称Greedy重数分配算法是最优于最大化上述期望的。
2. 分配重数时的参数s，不会限制重数具体值的上界：

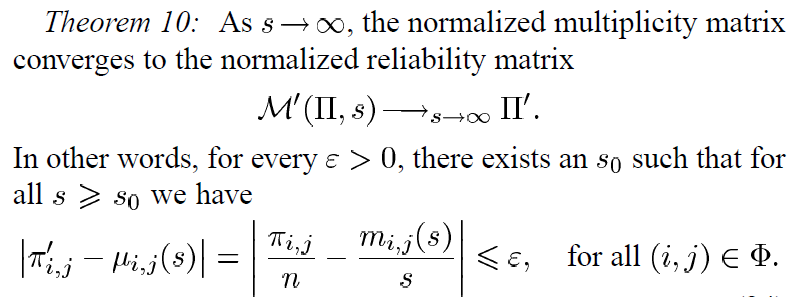


1. 有以下关系，用意不明，Gross, Warren J., et al. "Applications of algebraic soft-decision decoding of Reed-Solomon codes." IEEE transactions on communications 54.7 (2006): 1224-1234. 的图2有与此相关的讨论（他在讨论怎么选择此重数分配参数，但KV原文并不是为了提供这一证据）：

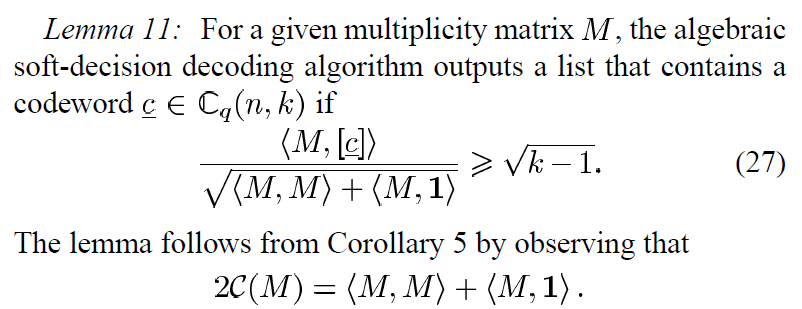


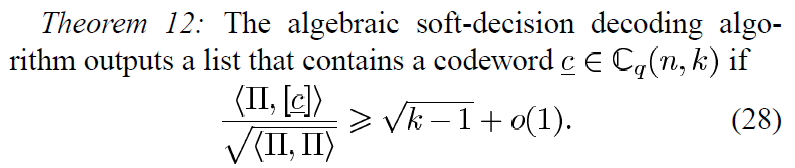


1. 重数分配代价参数趋于无穷时，归一化可靠度与归一化重数（分别用cost和码长做了归一化）是一致的：

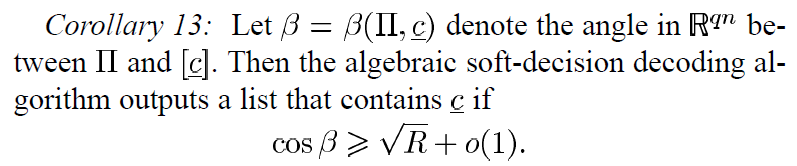


1. 重数分配问题被作者比作赌博问题，即有限的筹码（分配参数s或cost），有限的信息（信道可靠度矩阵），最大化回报（列表译码成功率）。
2. 以下定理描述了当cost趋于无穷时，本文算法的限制：

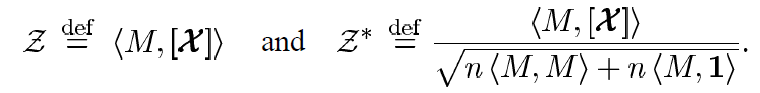




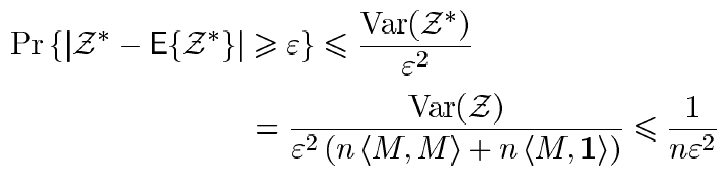
其几何表达为：



1. 文中第5节A的最后，有关于KV、GS、BM的译码区域几何描述。
2. 定义两个随机变量（因发射码字随机）：



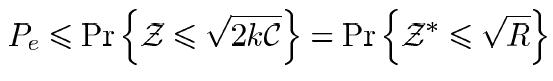
有形如切比雪夫不等式（弱大数定律）的关系：



则此Z\*是收敛于其期望的。

（20）可建立Z及Z\*与译码失败概率之间的关系：





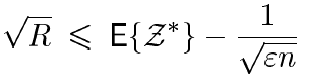
此不等式左半部分来源于：



注意，这是充分条件。

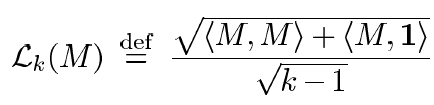
右半部分来源于Z与Z\*的定义。

1. 可靠传输的容量界为：



右边减去的项不知从何而来。

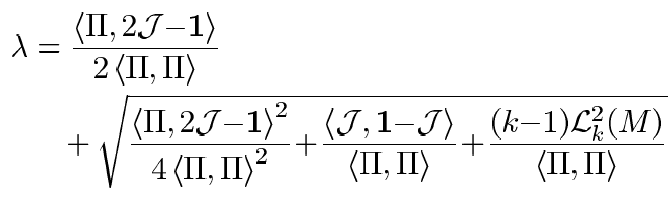
1. 列表译码输出码字的上限：



1. 重数分配有快速转换方法：



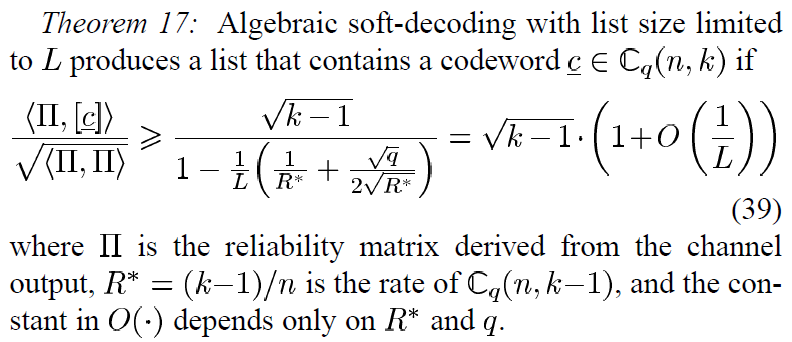
λ的表达式为：



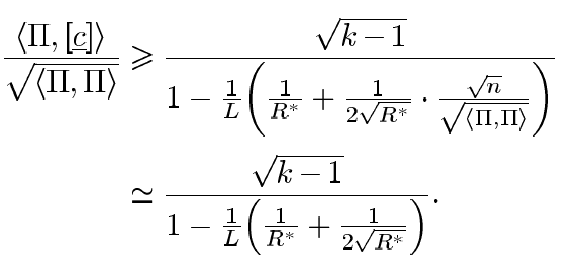
1. 如果用此算法时，想在分配重数时精确控制列表规模L，那么可以在分配重数时实时计算（21）中的公式，使得：



1. 本列表译码算法的输出包含特定码字的条件为：

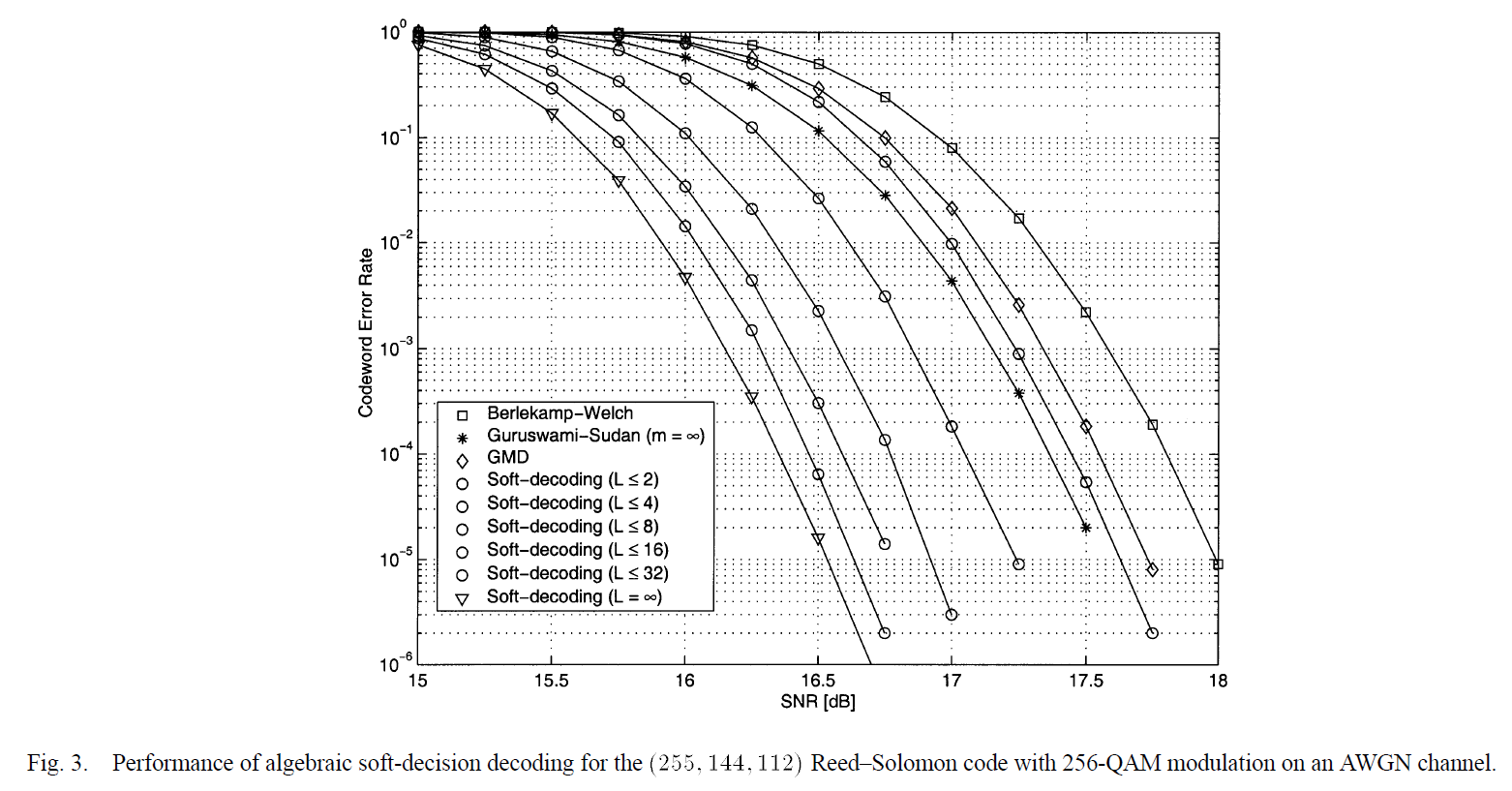


但此界较松，有更紧一点的界：

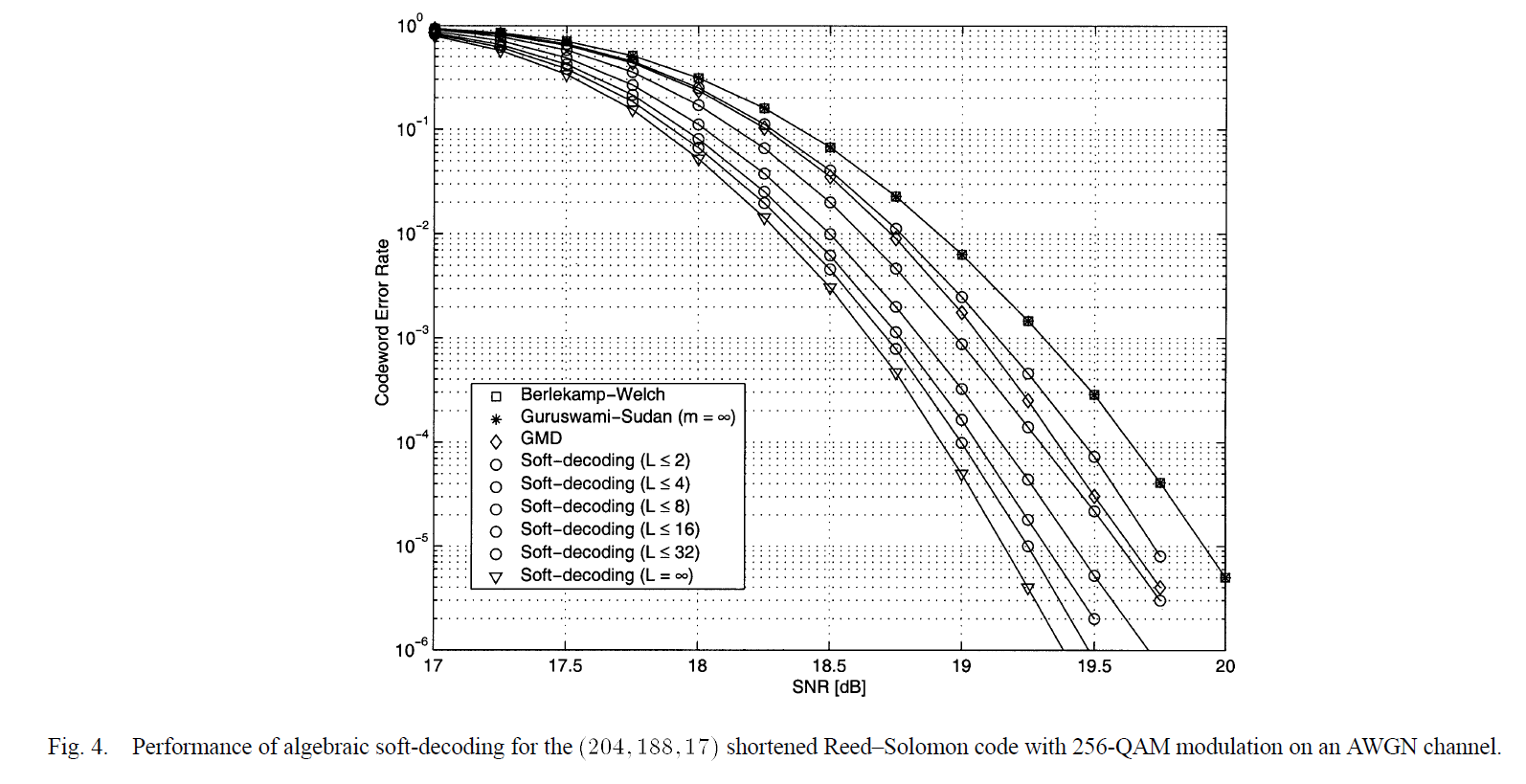


上述界表明，此列表译码性能以O(1/L)收敛，且要达到渐进性能的列表规模L是与码长n无关的。

1. 下图表明L=4就能获得不错的性能：



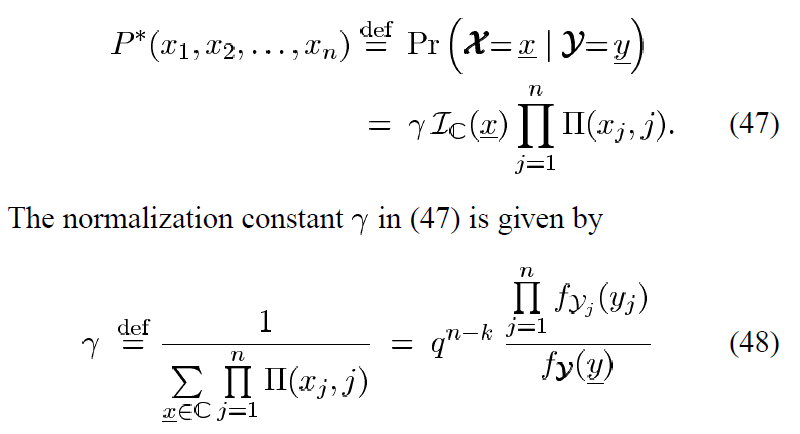
1. 下图表明高码率时软译码增益下降，但原因是列表译码的增益下降（L=4的性能还是可以的）：



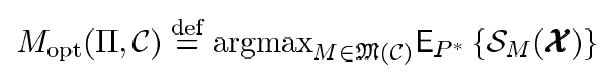
1. 如果是有记忆信道或级联码应用时，要用2N维的概率分布，2N的即N×N，即N个X（前序）影响N个可选输出符号（典型场景即1D ISI信道）。注意，如果是存储时，情况会变为2D ISI信道，这时，输出不光受N个前序影响，还会受同时输出（存储）的其他符号影响。

对于典型的有记忆信道的科普，可以看Zhong, Libo, Fady Alajaji, and Glen Takahara. "A binary communication channel with memory based on a finite queue." IEEE transactions on information theory 53.8 (2007): 2815-2840. 对于有记忆信道的描述，可以看Shental, Ori, et al. "Discrete-input two-dimensional Gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics." IEEE Transactions on Information Theory 54.4 (2008): 1500-1513.的第二节。

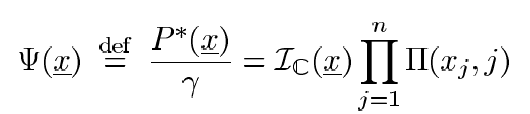
1. 有记忆信道的情况下，无记忆信道时的所有结论都依然适用。
2. 计算有记忆信道时的联合后验概率可用MAP算法，若是有限状态的可用BCJR算法（可将有记忆转为无记忆）。这些算法另开文档做记录与阅读。
3. 衰落信道或其他特别的信道下，如果译码器获得的信道信息准确，可以得到比AWGN无记忆信道下更多的编码增益。直观的理解应是这时得到的信道信息辅助更多了。
4. 实际上，计算后验概率时，还应将码本考虑在内，原因是发送符号并不是均匀分布于空间内的（因为编码后分布可能有变）：

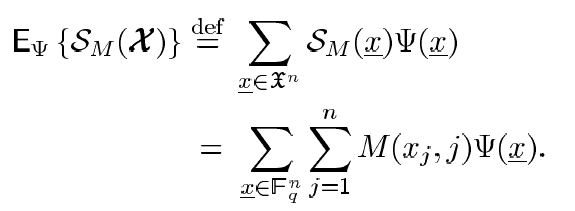


此时，优化问题变为：

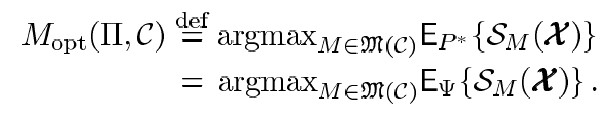


1. 计算是一个NP难问题，但是有：



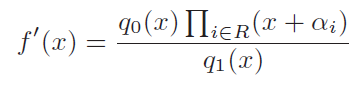


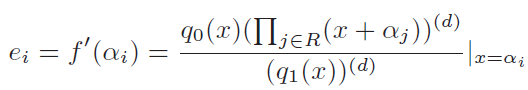
从而优化问题可以变为：



此外，码本的估计也不容易。

1. 优化重数分配问题实际上是NP难问题。这与RS码的最大似然译码有关系。具体的原因较为复杂。实际上，解决重数分配问题有后续的其他方法，如高斯法、切诺夫界法、灌水法等。详见黄勤硕士论文第4.4节的综述部分。
2. 注意，在用Factorization-free方法时，有以下计算方式（与赋值编码方法类似，重编码后R对应的码字为0，若赋值后不为0则必定叠加了错误）：





这一计算要用到洛必达法则，而洛必达法则的适用范围是0/0型或∞/∞型极限，此式并不一定满足此二者，不能直接对所有接收符号都套用洛必达法则，需分情况讨论，只有0/0型时才能使用洛必达法则。计算时，先对分母的q1(x)进行chien-search求根。求根后，即可知：

1. 使用洛必达法则的是同时为v(x)和q1(x)的根的本原元。
2. 若为v(x)的根不是q(x)的根，则此式结果为0。

Zhu, Jiangli, and Xinmiao Zhang. "Factorization-free low-complexity Chase soft-decision decoding of Reed-Solomon codes." 2009 IEEE International Symposium on Circuits and Systems. IEEE, 2009.中证明q0(x)不含v(x)的根，因此考虑0/0型时，不必去看q0(x)。

注意，这一方法只对R（可靠组）中的本原元进行，先恢复R，再用erasure decoding恢复整个码字。