Abstract

Greedy Algorithms in Coding Theory

Dejan Spasov

Institute of Informatics
Ss Cyril and Methodius University
Skopje, Macedonia

It is well known that a simple greedy algorithm will produce a code with very good code parameters. In this talk, we will discuss the space and time complexity of several greedy algorithms. We will start with the well-known Varshamov's algorithm and present two variations: memoryless Varshamov and exp-memory Varshamov. Both algorithms generate the parity check matrix H of a linear code. The memoryless Varshamov has space complexity $O\left(n^2\right)$ and time complexity

$$O\left(poly\left(n\right)q^{n(1-R+H(\delta))}\right),$$

where n is the code length, R is the code rate, δ is the relative distance, and $H\left(\delta\right)$ is the entropy function. On the other hand, exp-memory Varshamov is algorithm with exponential space complexity $\Theta\left(2^{\log(q)(1-R)n}\right)$ and improved, but still exponential, time complexity

$$O\left(poly\left(n\right)q^{n\max(1-R,H(\delta))}\right).$$

Next, we will present two relatively new algorithms that are focused on finding the A-matrix of a linear systematic code, $G = \begin{bmatrix} I & A \end{bmatrix}$. The Jenkins' algorithm builds the A-matrix with acceptable space complexity $O\left(n^2\right)$ and worst-case time complexity

$$O\left(poly\left(n\right)q^{n\left(1-\left(R+RH\left(\frac{\delta}{R}\right)\right)U(R-2\delta)\right)}\right),$$

where $U\left(R-2\delta\right)$ is the unit step function. In similar fashion, there exists a exp-memory Jenkins algorithm, known as the Lexicographic Construction, that has improved time complexity $O\left(poly\left(n\right)q^{n(1-R)}\right)$ and exponential space complexity

$$\begin{cases} \Theta\left(\log\left(n\right)2^{\log(q)(1-R)n}\right) & \delta \to const \\ \Theta\left(2^{\log(q)(1-R)n}\right) & \delta \to 0 \end{cases}$$

We will finish the talk with discussion of possible practical applications of the greedy algorithms. For example, we can combine the Lexicographic Construction and the Jenkins algorithm in order to improve the M. Grasls tables of best known linear codes.