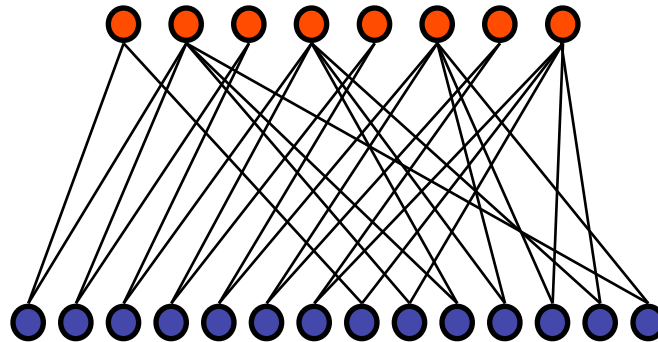


Verification Decoding of Raptor Codes

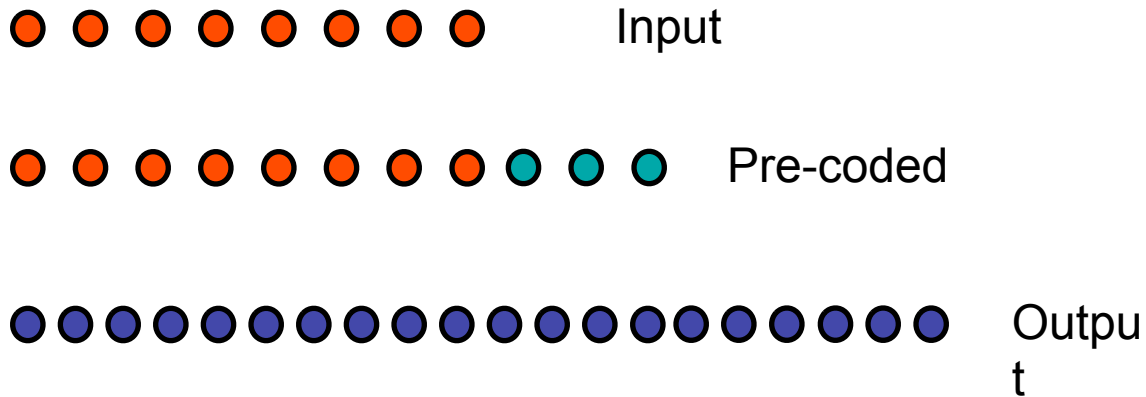


Amin Shokrollahi
Laboratoire d'algorithmique
Laboratoire de mathématiques algorithmiques
EPFL

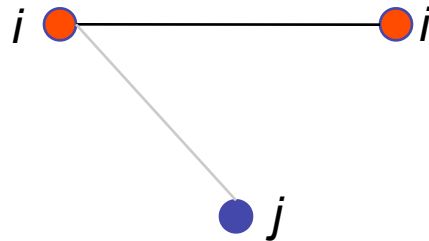
Joint work with R. Karp and M. Luby

Raptor Codes

Considering Raptor Codes of type $(k, \Omega(x), C)$: k is the number of input symbols, $\Omega(x)$ the degree distribution, and C is the pre-code.



q-ary Symmetric Channel



$$\text{Capacity} \sim 1 - p$$

Want to code on this channel and get arbitrarily close to the capacity.

Possible Solutions

Can use a part of every symbol as a hash, detect symbols for which hash doesn't match, and reduce to erasure coding.

However, error probability would not be very good.

Will use adaptation of “verification decoding” instead.

Method was introduced by Luby and Mitzenmacher for decoding of LDPC codes on the q -ary symmetric channel.

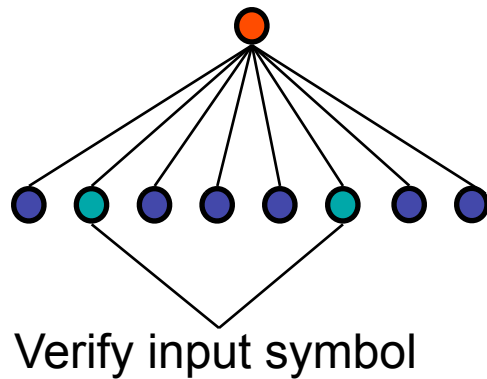
We will assume that q is very large in the following. The error probabilities will always be at least of the order of $1/q$.

Simple Algorithm

For every input symbol if two neighbors are of degree one and are equal, decode the input symbol to that common value.

Add value of input symbol to neighbors, and remove input symbol from graph.

Continue.



What is the Best we can Expect?

Every input symbol needs two correctly transmitted output symbols for recovery.

So, number of correctly transmitted output symbols has to be at least twice the number of input symbols.

Capacity result says that we should be able to correct if number of correctly transmitted symbols is roughly equal to number of input symbols.

So, we can correct only at most to half of the “capacity”.
But can we achieve this?

Message Passing Formulation

Input symbol v

Output symbol c

Message space: $GF(q) \cup \{E\}$

$$m_{cv} = \begin{cases} E, & \text{if } \exists v' \neq v: m'_{v'c} = E, \\ \sum_{v' \neq v} m'_{v'c}, & \text{else,} \end{cases}$$

$$m'_{vc} = \begin{cases} m, & \text{if } \exists c' \neq c'', c', c'' \neq c: \\ & m_{c'v} = m_{c''v} = m, \\ E, & \text{else.} \end{cases}$$

Message Passing Formulation

$$p_i = \Pr[m_{vc}^{(i)} = \mathbf{E}]$$

α Average degree of input symbols

$$\omega(x) = \frac{\Omega'(x)}{\Omega'(1)}$$

$$p_{i+1} = e^{-\alpha\omega(1-p_i)}(1 + \alpha\omega(1 - p_i))$$

Want:

$$x < 1 - e^{-\alpha\omega(x)}(1 + \alpha\omega(x))$$

for $0 < x < 1 - \epsilon$.

A Little Theorem

Suppose that $g(x)$ is a function such that $g(x), g'(x), g''(x) > 0$ for $x \in (0, 1)$, and such that

$$x < 1 - e^{-g(x)}(1 + g(x))$$

for $0 < x < 1 - \epsilon$ for some $\epsilon > 0$. Then we have

$$\int_0^1 g(x) dx \geq 3 + \frac{1}{e} - \frac{e}{2} - f(\epsilon)$$

for some continuous monotonically increasing function $f(x)$ with $f(0) = 0$.

And its Corollary

Overhead of the algorithm: How much more output symbols are needed than the minimum necessary as a fraction of the number of input symbols.

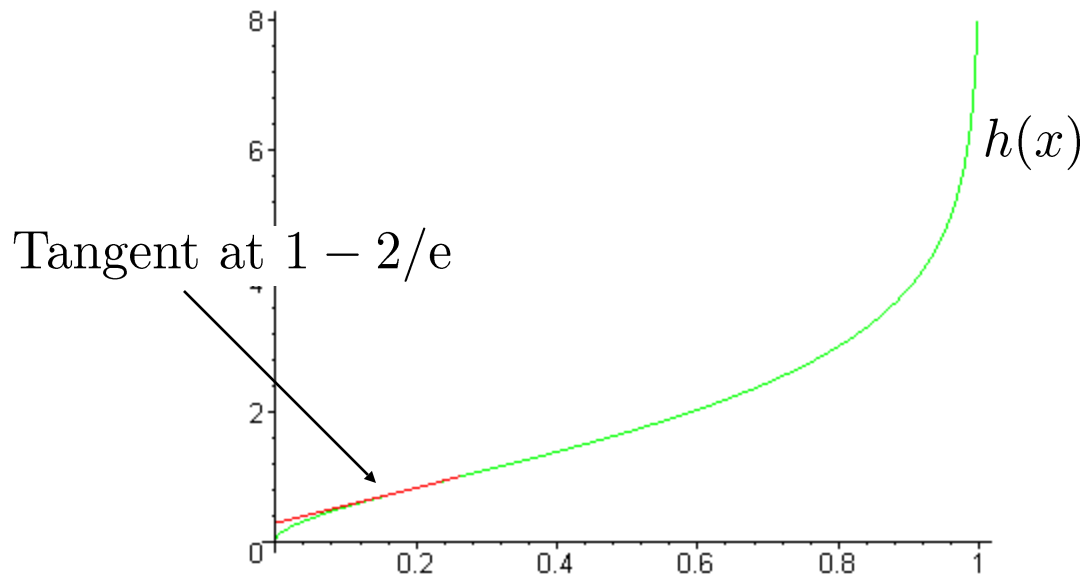
$$\text{Overhead} = \int_0^1 \alpha \omega(x) dx - 1$$

The overhead of the simple algorithm is at least $2 + 1/e - e/2 - f(\epsilon) \simeq 1.00873 - f(\epsilon)$.

(Sketch of the) Proof of the Theorem

The function $1 - e^{-x}(1 + x)$ has an inverse $h(x)$ on $[0, 1]$.

$g(x) := \alpha\omega(x)$ must be always larger than $h(x)$, for $0 < x < 1 - \epsilon$.



Raptor Code

This would take care of the LT-code. What about the Raptor code?

At the end of the LT-decoding the input symbols that are not yet determined are regarded as erasures.

Pre-code needs to clean up erasures only. Total overhead:

$$\frac{1}{x_0} \left(3 + \frac{1}{e} - \frac{e}{2} - \int_{h(x_0)}^{\infty} y^2 e^{-y} dy + (1 - x_0)h(x_0) \right) - 1$$

where

$$x_0 = 1 - \epsilon$$

Good Degree Distributions

$$\begin{aligned}
 g(x) = & 0.283927 + 2.704530x + 1.190185x^5 + 0.508412x^6 \\
 & + 0.441844x^{17} + 0.778682x^{18} + 1.283476x^{48} \\
 & + 13.709022x^{299}.
 \end{aligned}$$

gives overhead of 2.044606.

