ARQ with Doped Fountain Decoding

Silvija Kokalj-Filipović, Predrag Spasojević, Emina Soljanin, and Roy Yates

System Model

- (Distributed) Source encodes k independent packets into n>k Fountain codesymbols
- Codesymbol degrees follow Ideal Soliton distribution
- Destination collects random set of k codesymbols defining the columns of the Generator Matrix (GM)
- Destination begins Belief Propagation (BP) Decoding iff the GM contains a degree-one column

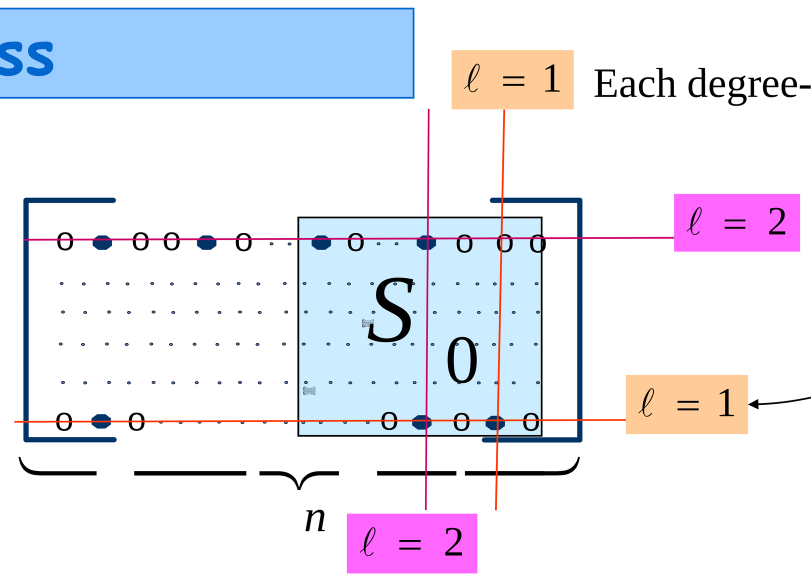
Decoding Process

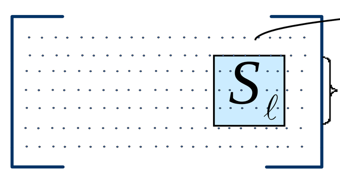
Start:

$$S_0 = \{s_{ij}\}_{k\times k}$$

After ℓ rounds:

$$S_\ell = \{s_{ij}\}_{(k-\ell)\times(k-\ell)}$$

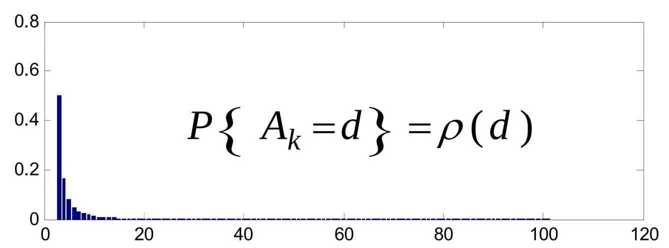




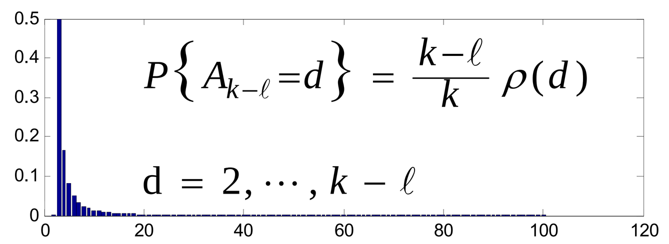
Column Degree $A_{k-\ell}$: number of ones in a random column of S_ℓ

Degree Evolution

- Start with Ideal Soliton Distribution



- Support of output symbol degrees after ℓ processed input symbols is $\{1,\cdots,k-\ell\}$



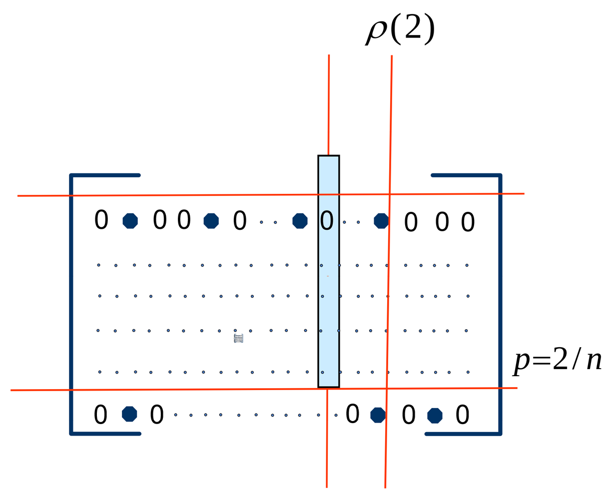
Distribution shape preserved within new support

- **Ripple**: Set of degree-one code symbols (columns)

Ripple is what we know and use to learn what we don't know (higher-degree symbols)

BP as a Peeling Decoder

- Layer by layer of coded symbols (and their degrees) peeled by the ripple
- Peeled degree-two symbols become Ripple
- Degree-two symbols are Important!!!



Ripple Evolution

Ripple generated from columns of degree two

Distribution shape preserved => $\Pr\{A_{k-\ell}=2\} = \rho(2) = \frac{1}{2}$ for all ℓ

Doping with random symbol would generate ripple increment of size R

$$R \sim Bi\left(n\rho(2), \frac{2}{n}\right)$$

The same stochastic description regardless of the doping round ℓ

$$R \sim Bi\left(\frac{n}{2}, \frac{2}{n}\right) \approx Poiss(1)$$

Analytical Model

Ripple increments create a sequence of IID RVs $R_i \sim Poiss(1)$

$$Poiss(1): \eta(\rho) = \frac{e^{-1}}{r!}$$

Ripple is a partial sum of IID RVs $S_n = \sum_{i=1}^{n} R_i$

and can be analyzed using random-walk (RW) based models

Each value of the RW is a state in the Markov Chain (MC) described by **P**

$$P = \begin{bmatrix} \eta(1) & \eta(2) & \eta(3) & \cdots & \eta(m) \\ \eta(0) & \eta(1) & \eta(2) & \cdots & \eta(m-1) \\ 0 & \eta(0) & \eta(1) & \cdots & \eta(m-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \eta(0) & \eta(1) \end{bmatrix}$$

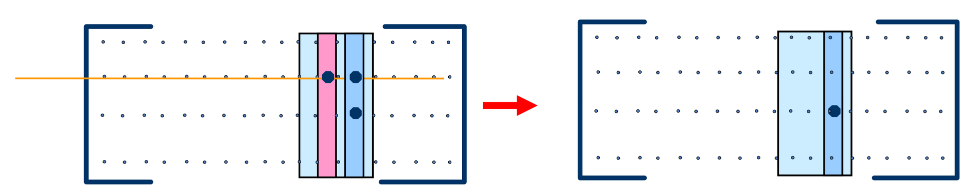
Stalled decoding event: EMPTY RIPPLE

$$P_t = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix} P^{t-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

P_t : probability of ripple becoming empty at time t

Unlocking Stalled Decoding Event

Doping:
Bring in a degree-one symbol that will peel-off a non-empty set of degree-two codesymbols (columns)
With doping, ripple evolution is a renewal process



Doping Cost Analysis

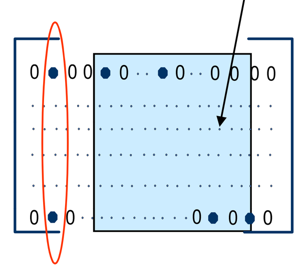
$$E[Y] = \sum tP_t$$

Ripple evolution indicates how many dopings $k_d \approx \frac{k}{E[Y]}$ we need

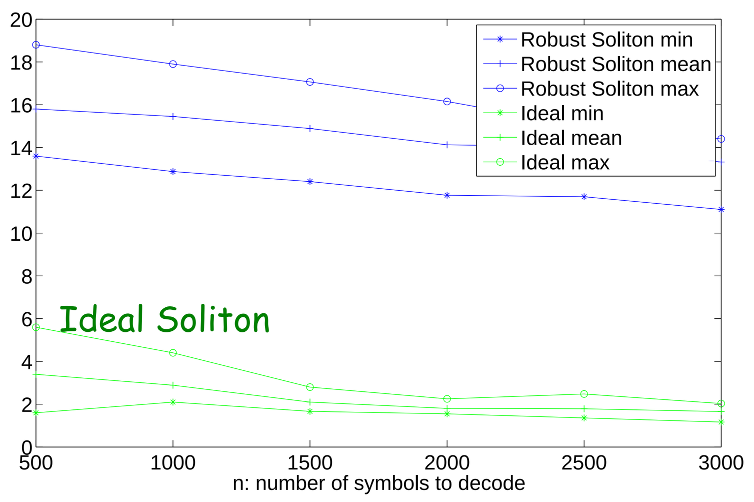
Interdoping yield Y: number of decoded symbols between dopings

Performance of Doped Fountain Decoder

Doping Cost $\frac{k_s + k_d - k}{k}$ Depends on Degree Distribution



$$S_0 = \{s_{ij}\}_{k\times k_s}$$



Doping Cost Depends on Doping Mechanism