Karlsruhe Institute of Technology

SPONSORED BY THE Federal Ministry of Education and Research

European Research Council
Established by the European Commission

Structural Optimization of Factor Graphs for Symbol Detection via Continuous Clustering and Machine Learning

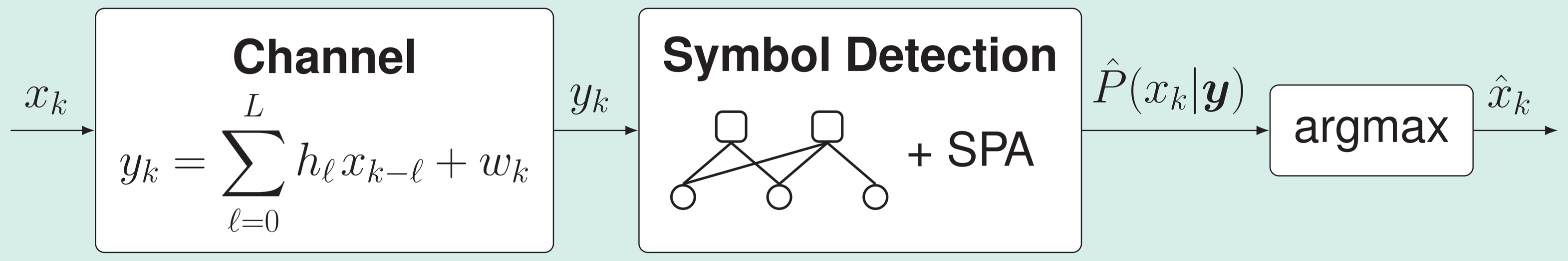
Lukas Rapp, Luca Schmid, Andrej Rode, and Laurent Schmalen

1. Introduction

- Inference tasks can be efficiently calculated via the sum-product algorithm (SPA) on a corresponding factor graph
- **Problem:** For factor graphs with cycles, the SPA performance heavily relies on the factor graph structure
- **Idea:** Optimize SPA performance by learning the factor graph structure

2. Symbol Detection

- **Example inference task:** Transmission of independent uniformly distributed BPSK symbols over an inter-symbol interference channel



- Impulse response: $\boldsymbol{h} = [0.407, 0.100, 0.815, 0.100, 0.407] \in \mathbb{R}^{L+1}$
- Additive white Gaussian noise (AWGN): $w_k \sim \mathcal{N}(0, \sigma^2)$

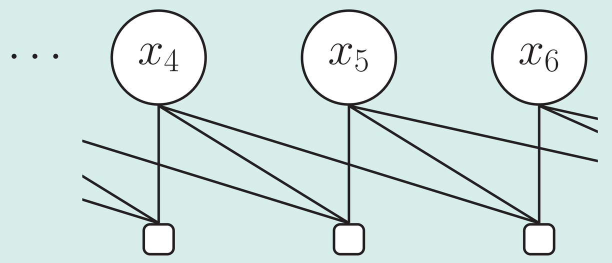
- Symbol detection via marginalization of $P(\boldsymbol{x} \mid \boldsymbol{y})$ using the SPA:

$$P(x_k \mid \boldsymbol{y}) = \sum_{\sim\{x_k\}} P(\boldsymbol{x} \mid \boldsymbol{y}), \quad P(\boldsymbol{x} \mid \boldsymbol{y}) \propto \prod_{k=1}^{K} \exp\left(-\frac{1}{2\sigma^2}\left|y_k - \sum_{\ell=0}^{L} h_\ell x_{k-\ell}\right|^2\right)$$

3. Factor Graph Models for Symbol Detection

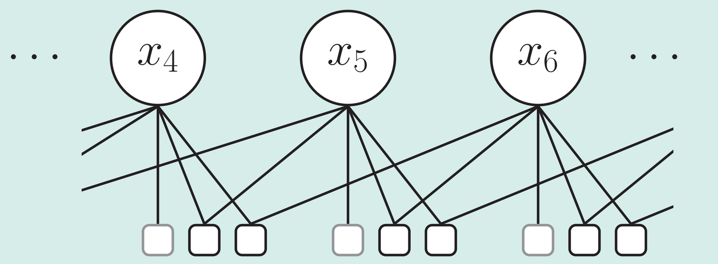
- **Problem:** SPA complexity increases exponentially with factor node (FN) degree
- Common factor graph models:

Forney Factor Graph (FFG)



− FN degree $L+1$
→ High complexity
+ Near-MAP performance

Ungerboeck Factor Graph (UFG)

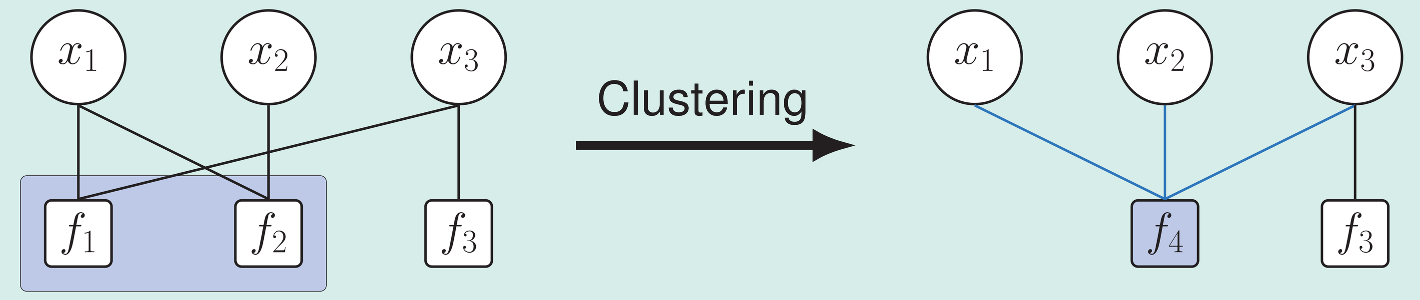


+ FN degree 1 and 2
→ Low complexity
− Poor performance

- **Idea:** Create new factor graphs by **clustering** FNs of the UFG
→ Interpolation between UFG and FFG

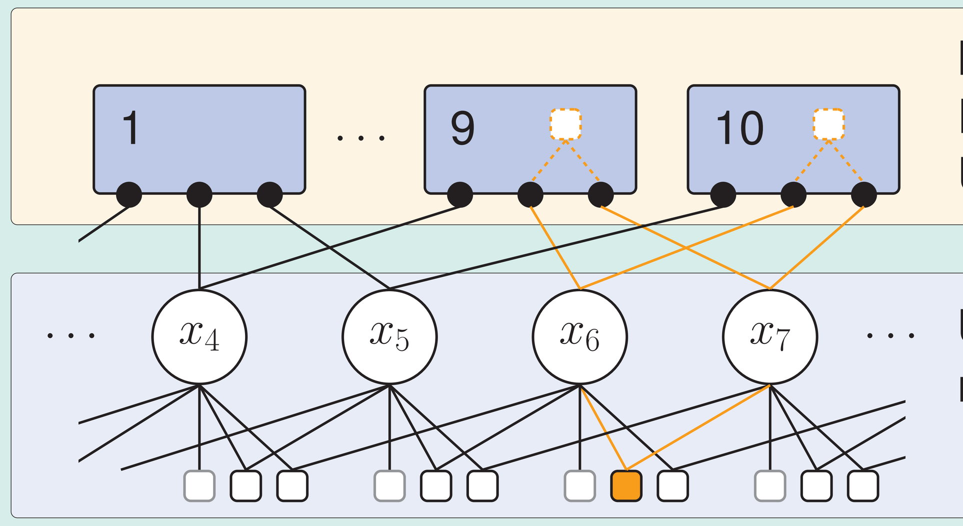
4. Factor Node Clustering

- Clustering of FNs: Combine FNs by multiplying their factors. Example:



$$f(x_1, x_2, x_3) = \underbrace{f_1(x_1, x_3) f_2(x_1, x_2)}_{f_4(x_1, x_2, x_3)} f_3(x_3)$$

- **Idea:** FN containers transform clustering problem into systematic form



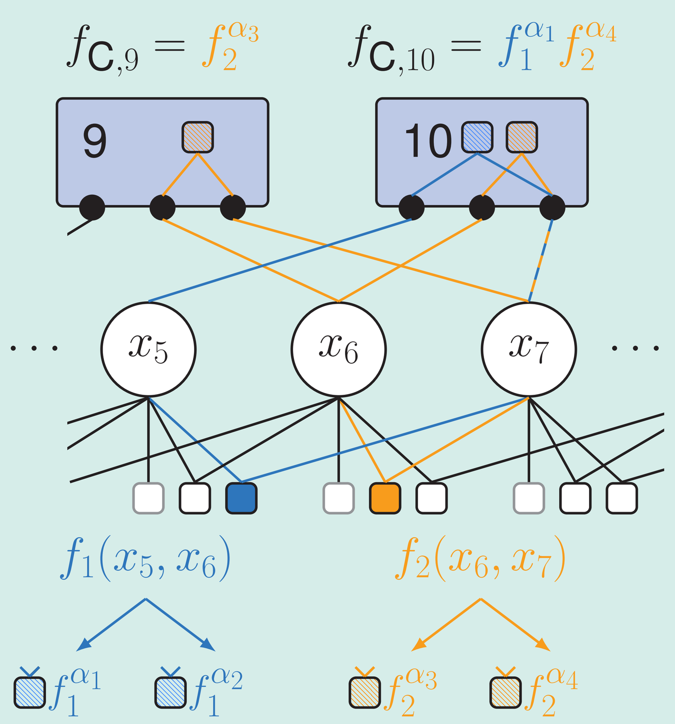
FN Containers: Empty FNs in which FNs of the UFG can be clustered

UFG: Basis to construct new FGs via clustering

- **Clustering options:** Each FN of the UFG can be clustered into the FN containers that are connected with the same variable nodes (VNs)
- Create new factor graphs by putting all FNs in one of their options
- FN container degree defines SPA complexity

5. Continuous Clustering (CC)

- **Idea:** Instead of clustering each FN in only one container, use every FN simultaneously in all of its clustering options:



Step 2: Cluster the factorized factors $f_i^{\alpha_j}$ into connected FN containers f_{C}

Step 1: Factorize all FNs f_i of the UFG via weights $\alpha_j \in [0, 1]$:

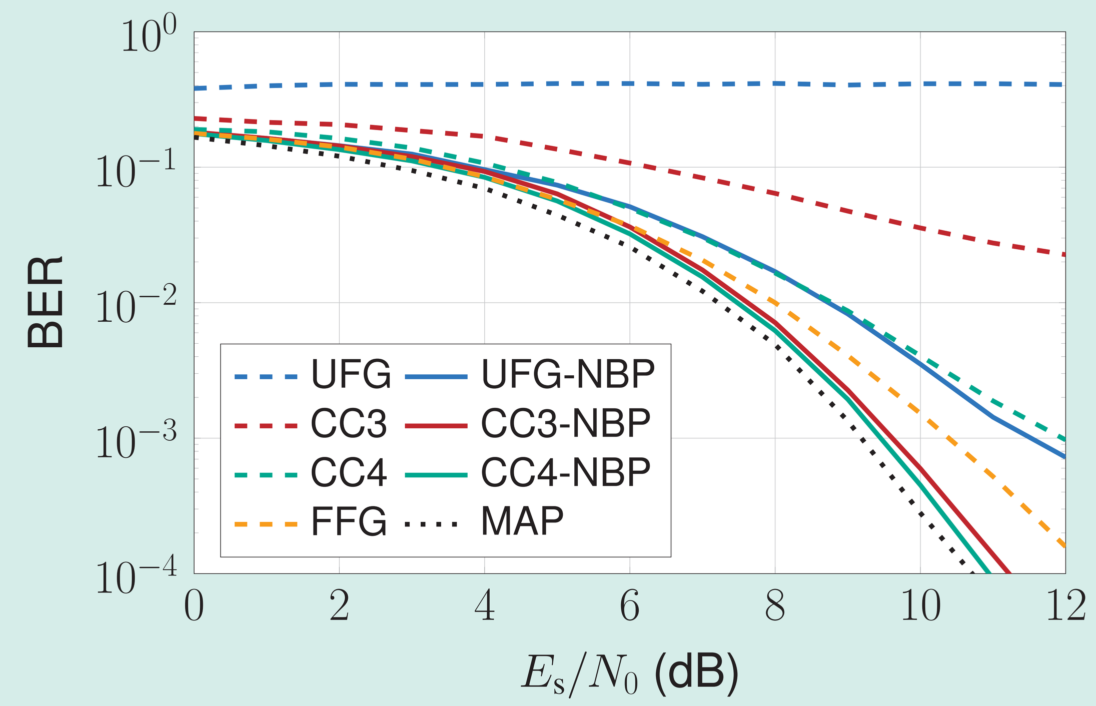
$$f_i(\cdot) = \prod_j f_i^{\alpha_j}(\cdot), \quad \sum_j \alpha_j = 1$$

6. Optimization

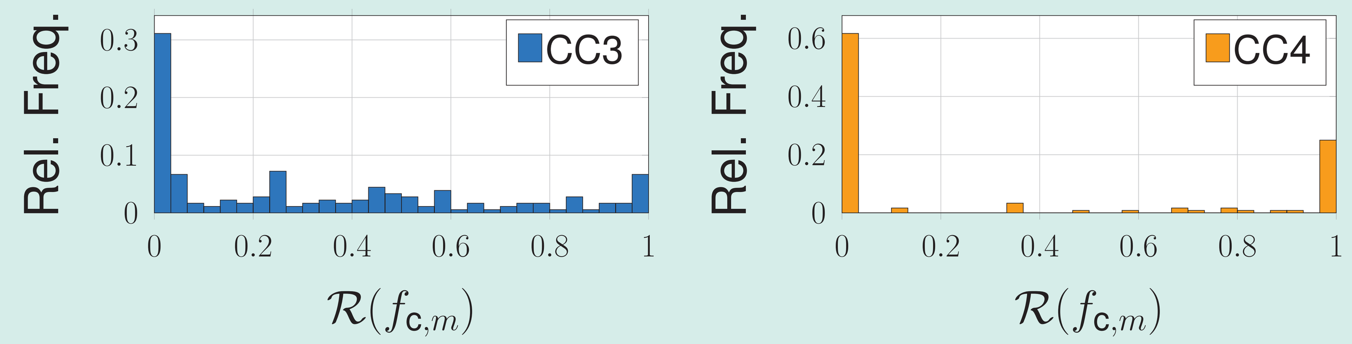
- Factor graph structure is parameterized by continuous weights α_{ij}
→ Enables optimization of SPA performance via **gradient descent**
- Direct combination with **neural belief propagation (NBP)** possible
- **Structure extraction:** Learned graph is extracted by pruning empty FN containers $f_{\mathsf{C},m}$ (i.e., $\mathcal{R}(f_{\mathsf{C},m}) \approx 0$)
  - Relevance $\mathcal{R}(f_{\mathsf{C},m})$: Maximum weight α_{ij} in container $f_{\mathsf{C},m}$

7. Results

- Bit error rate (BER) for factor graphs based on containers of degree 3 or 4 (CC3 or CC4) with and without NBP:

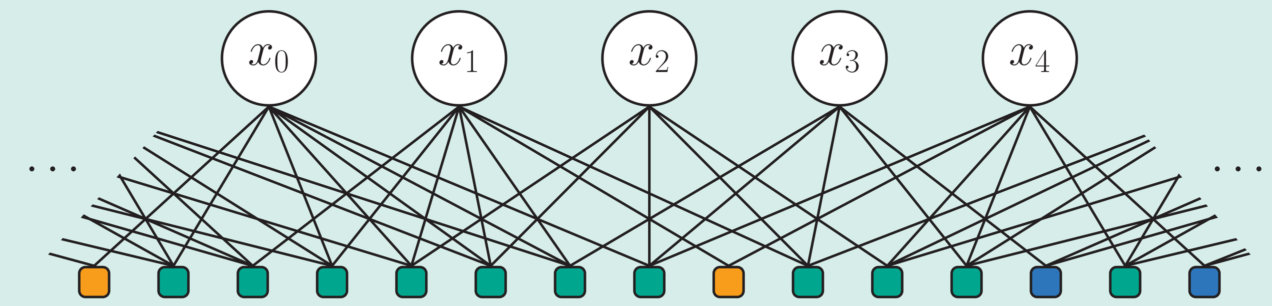


- Histogram of the relevance of the learned FN containers:



→ Large proportion of containers is irrelevant → Pruning possible

- Section of CC4 after pruning with $\alpha_{\text{thr}} = 0.01$:



FN Degree: 2 3 4