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: \( h = [0.407, 0.100, 0.815, 0.100, 0.407] \in \mathbb{R}^{L+1} \)
- Additive white Gaussian noise (AWGN): \( w_k \sim \mathcal{N}(0, \sigma^2) \)
- Symbol detection via marginalization of \( P(x | y) \) using the SPA:
  \[
  P(x_k | y) = \sum_{x_{k-1}} P(x_{k-1} | y) \cdot P(x_k | x_{k-1}) \cdot \exp \left( \frac{1}{2\sigma^2} \sum_{l=0}^{L} h_l x_{k-l} - y_k \right)
  \]

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)
  - Ungerboeck Factor Graph (UFG)
- Idea: Create new factor graphs by clustering FN containers into connected FN containers

4. Factor Node Clustering
- Clustering of FN: Combine FN by multiplying their factors. Example:
  \[
  f(x_1, x_2, x_3) = f_1(x_1, x_2) f_2(x_2, x_3) f_3(x_3)
  \]
- Idea: FN containers transform clustering problem into systematic form

5. Continuous Clustering (CC)
- Idea: Instead of clustering each FN in only one container, use every FN simultaneously in all of its clustering options:
  \[
  f_{c} = f_{C} \cdot f_{C^2} \cdot f_{C^3} \cdot f_{C^4}
  \]
- Step 1: Factorize all FN, \( f_i \), of the UFG via weights \( \alpha_j \in [0, 1] \):
  \[
  f_i(x_1, x_2) = \alpha_1 f_i^{(1)}(x_1, x_2) + \alpha_2 f_i^{(2)}(x_1, x_2) + \cdots + \alpha_5 f_i^{(5)}(x_1, x_2)
  \]
- Step 2: Cluster the factorized FN into connected FN containers \( f_C \)

6. Optimization
- Factor graph structure is parameterized by continuous weights \( \alpha_j \)
  \( \rightarrow \) 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 \( f_{c_{\alpha}} \) (i.e., \( R(f_{c_{\alpha}}) \approx 0 \))
  \( \rightarrow \) Relevance \( R(f_{c_{\alpha}}) \): Maximum weight \( \alpha_j \) in container \( f_{c_{\alpha}} \)

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: