Linear Programming-based Decoding of Turbo-Like Codes and its Relation to Iterative Approaches

Jon Feldman
jonfeld@theory.lcs.mit.edu
MIT LCS

David Karger
karger@theory.lcs.mit.edu
MIT LCS

Martin Wainwright
martinw@eecs.berkeley.edu
UC Berkeley

.

Decoding via Linear Programming

- New algorithm for decoding any turbo-like code [Feldman, Karger, FOCS 2002].
- Uses *linear programming (LP) relaxation*.
- Precise characterization of noise patterns that cause decoding error for BSC, AWGN: “noisy promenades.”
  - Reminiscent of work on “stopping sets” in the BEC [Di, Proietti, Richardson, Telatar, Urbanke, ’02; Richardson, Urbanke, Allerton’02].
- For rate-1/2 Repeat-Accumulate (RA) codes:
  - $\mathrm{WER} \leq n^{-\epsilon}$, $\quad (\mathrm{noise} < f(\epsilon))$.
- ML certificate property:
  - Outputs ML information word, or “error.”

Our Contributions

- Iterative subgradient decoding for any turbo-like code:
  - Uses trellis passes, message-passing.
  - $\exists$ step size guaranteeing convergence to same solution as LP decoder.

    $\rightarrow$ Same noise pattern error conditions, WER bounds.

    $\rightarrow$ ML certificate property.

Our Contributions

- Iterative subgradient decoding for any turbo-like code:
  - Uses trellis passes, message-passing.
  - $\exists$ step size guaranteeing convergence to same solution as LP decoder.

    $\rightarrow$ Same noise pattern error conditions, WER bounds.

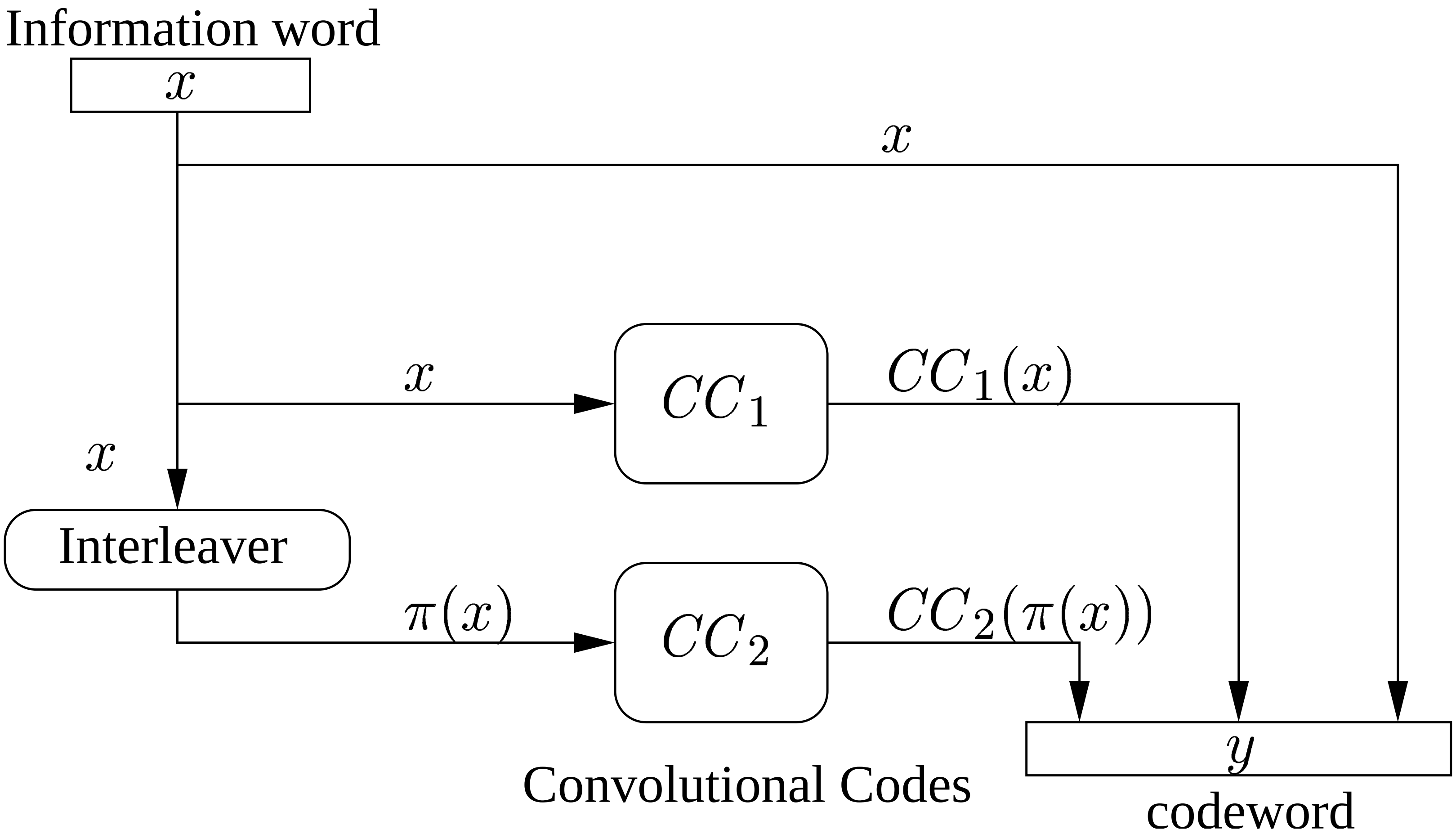
    $\rightarrow$ ML certificate property.
- Relation to Tree-Reweighted Max-Product (TRMP):
  - Iterative algorithm for finding optimal configurations on factor graphs [Wainwright, Jaakkola, Willsky, Allerton'02], Session V.B, Friday, 10am.
  - Turbo-like codes: simple message-passing decoder.
  - If constituent codes agree on a code word, it is the ML code word.

Outline

1. Turbo Codes, RA codes.
2. LP-based decoding of turbo-like codes.
3. Lagrangian dual form of LP:
   - Subgradient decoding,
   - TRMP decoding.
4. Noisy Promenades.

Classic Turbo Codes

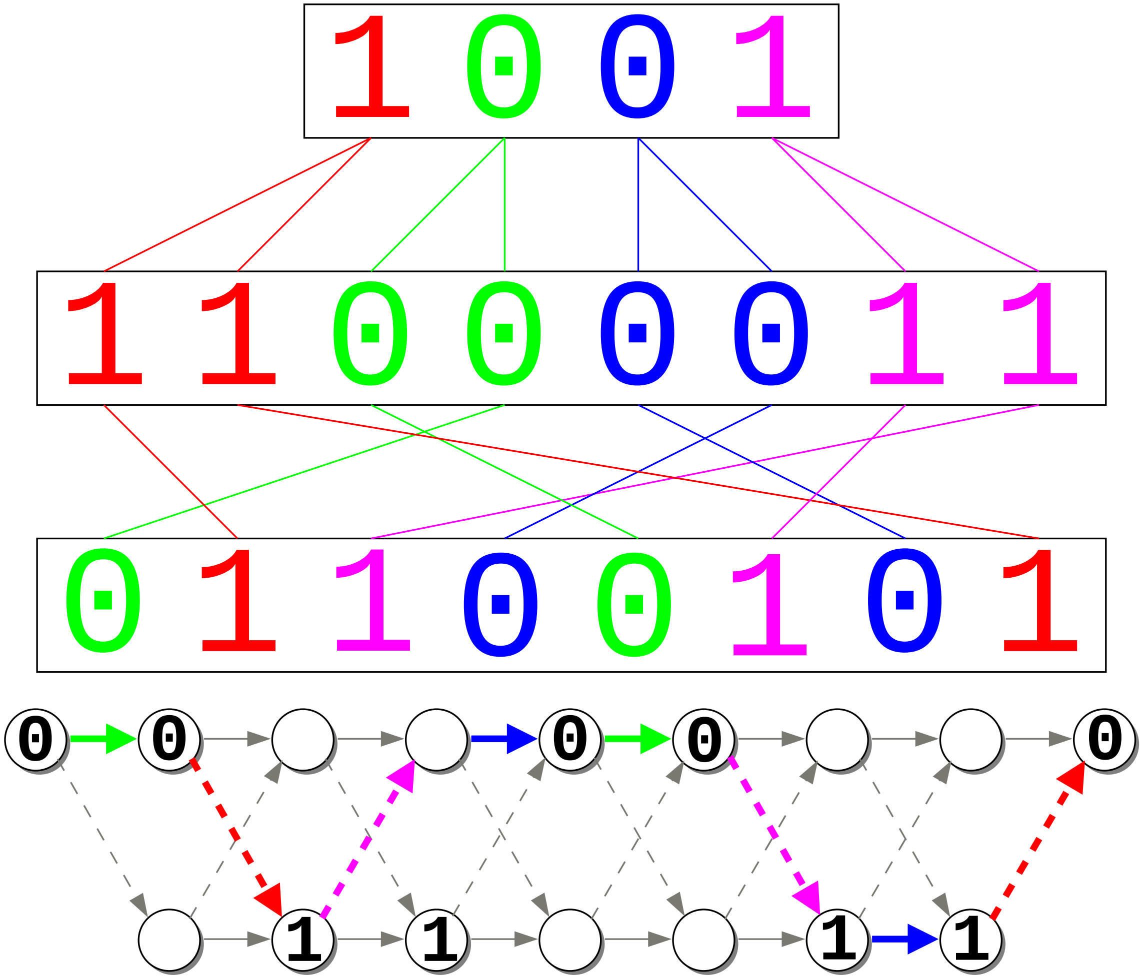
[Berrou, Glavieux, Thitimajshima, 1993]



- “Turbo-like” codes: Parallel, serial concatenated convolutional codes.

Repeat-Accumulate Codes

[Divsalar, Jin, McEliece, 1998]



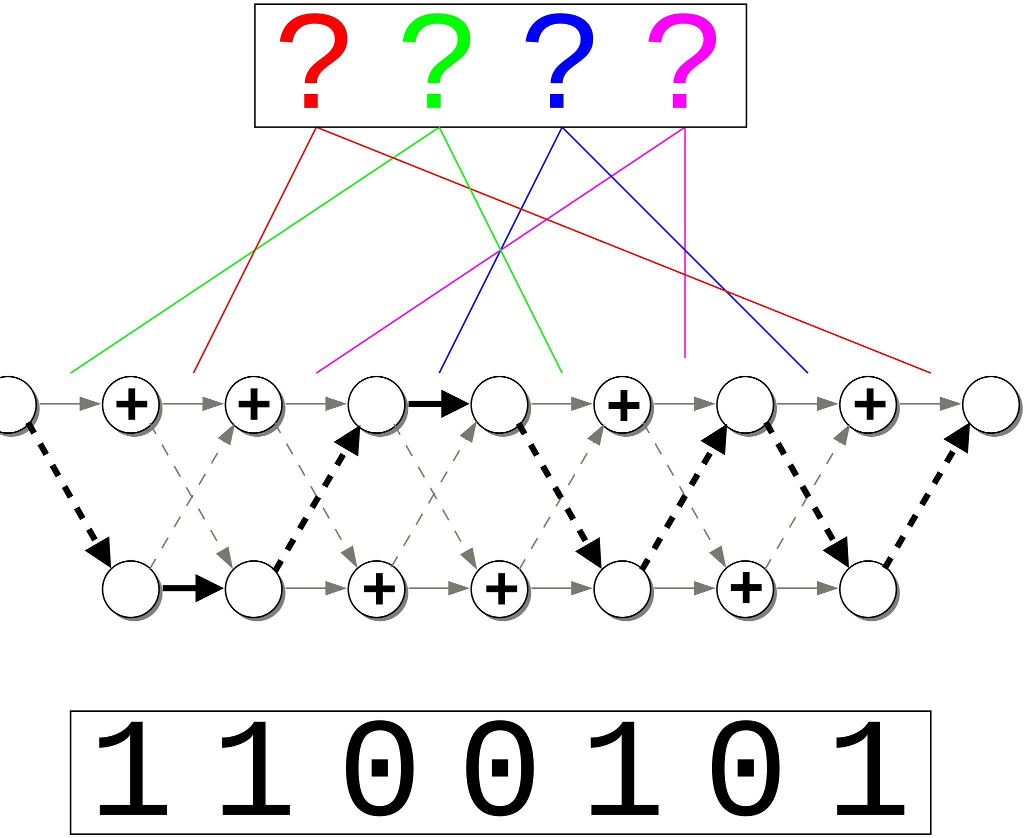
Information (word x)

Repeat

Permute

Accumulate (path $P(x)$)

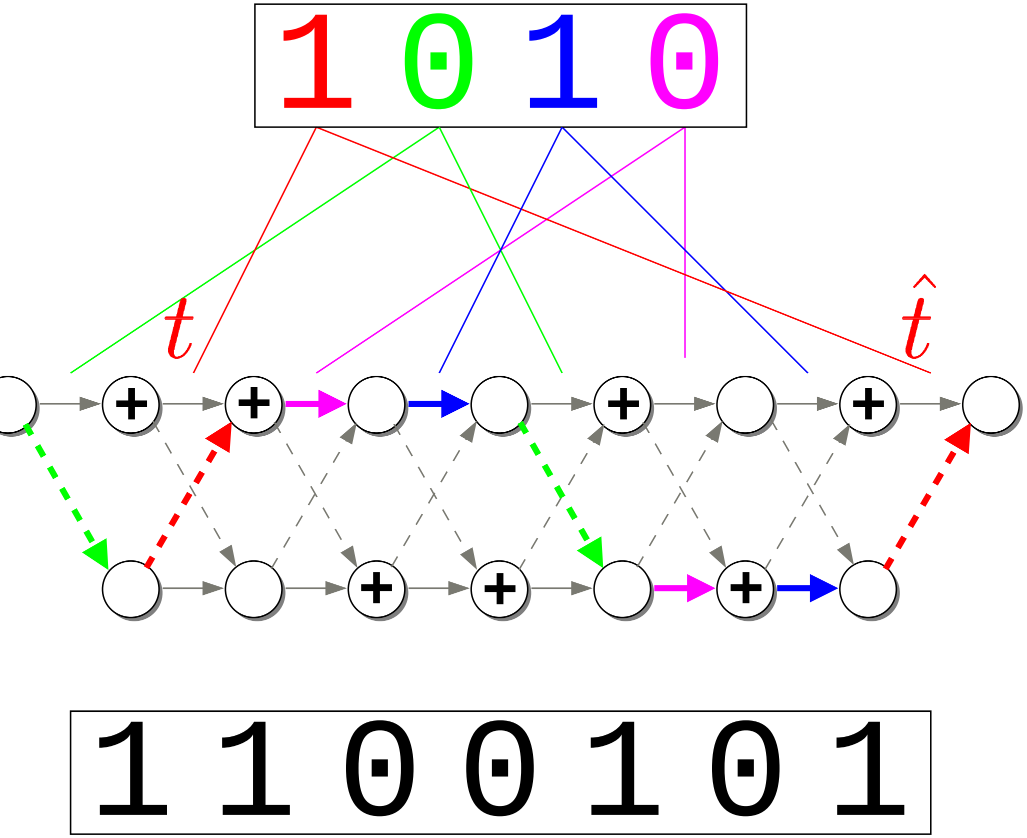
Decoding



Received word

- Costs on nodes: local log-likelihood ratio (LLR).
- Viterbi algorithm: finds max-cost path.
- Max-cost path does not necessarily correspond to code word.

Max-Likelihood Agreeable Path



Received word

- Path P is *Agreeable* if, for all info bits x_i:

  "1-edge" at t and $\hat{t}$, or

  "0-edge" at t and $\hat{t}$

- How do we find ML agreeable path?

Turbo Code Linear Program

- Variable f_P for all paths P, $0 \le f_P \le 1$. Cost c_P = $\sum_{e \in P} c_e$.

  For rate-1/2 RA codes (RALP):

$$
\begin{array}{lrcl}
\max & \sum_P c_P f_P & & \text{s.t.} \\
& \sum_P f_P & = & 1 \\
\forall x_i,\ X_i = \{t, \hat{t}\}, & \sum_{P \in S(t)} f_P & = & \sum_{P \in S(\hat{t})} f_P
\end{array}
$$

- $S(t)$: set of paths that “switch” at segment t.
- $X_i = \{t, \hat{t}\}$: two copies of x_i.
- Natural generalization for any turbo-like code.

Using RALP to Decode

- Solving RALP finds maximum-likelihood *agreeable distribution* f^* on paths.
- Strict "relaxation" of ML decoding problem.
- All the mass on one path: "integral solution."

Using RALP to Decode

- Solving RALP finds maximum-likelihood *agreeable distribution* f^* on paths.
- Strict "relaxation" of ML decoding problem.
- All the mass on one path: "integral solution."
- If f^* integral:
  - $f^*_P = 1$ for some P.
  - $f^*_{P'} = 0$ for all $P \neq P'$.
  - P is the ML agreeable path.
- If not, f^* is an agreeable *convex combination* of paths.
  - Output "error."

Solving RALP

- Use generic LP solver.
  - Ellipsoid algorithm: provably poly-time, but impractical.
  - Simplex algorithm: useful in practice, but not in real time.

Solving RALP

- Use generic LP solver.
  - Ellipsoid algorithm: provably poly-time, but impractical.
  - Simplex algorithm: useful in practice, but not in real time.
- Solve using subgradient algorithm:
  - Operates on *Lagrangian dual* form of the LP.
  - Takes the form of a standard message passing decoder

Lagrangian Dual

- Lagrange multipliers λ_i for each info bit x_i.
- For a path P, cost under λ:

$$\mathcal{L}(P, \lambda) = c_P + \sum_{x_i} \lambda_i A_i(P)$$

"agreeability" $A_i = \begin{cases} +1 & \text{if } P \in S(t)\ P \notin S(\hat{t}) \\ 0 & \text{if } P \text{ agreeable for } x_i \\ -1 & \text{if } P \notin S(t)\ P \in S(\hat{t}) \end{cases}$

Lagrangian Dual

- Lagrange multipliers λ_i for each info bit x_i.
- For a path P, cost under λ:

$$\mathcal{L}(P, \lambda) = c_P + \sum_{x_i} \lambda_i A_i(P)$$

"agreeability" $A_i = \begin{cases} +1 & \text{if } P \in S(t)\ P \notin S(\hat{t}) \\ 0 & \text{if } P \text{ agreeable for } x_i \\ -1 & \text{if } P \notin S(t)\ P \in S(\hat{t}) \end{cases}$

- Cost λ_i on 1-edges at segment t.
- Cost $-\lambda_i$ on 1-edges at segment $\hat{t}$.
- Natural generalization to any parallel concatenated convolutional code.

Lagrangian Dual, continued...

- Dual function $Q(\lambda) = \max_{P} \{\mathcal{L}(P, \lambda)\}$.
- Let $\hat{P}(\lambda) = \arg\max_{P} \{\mathcal{L}(P, \lambda)\}$.

Lagrangian Dual, continued...

- Dual function $Q(\lambda) = \max_P \{\mathcal{L}(P, \lambda)\}$.
- Let $\hat{P}(\lambda) = \arg\max_P \{\mathcal{L}(P, \lambda)\}$.
- Let $\lambda^* = \arg\min_\lambda Q(\lambda)$. By LP duality, $\sum_P c_P f_P^*$ = $Q(\lambda^*)$.
- Find λ^* using sequence of “message-passing” updates:

$$\lambda^{m+1} = \lambda^m - \alpha^m A_i\big(\hat{P}(\lambda^m)\big)$$

- Subgradient $A_i\big(\hat{P}(\lambda^m)\big)$ computed w/ Viterbi algorithm.
- Appropriate step size α^m assures convergence to λ^*.

Lagrangian Dual, continued...

- Dual function $Q(\lambda) = \max_P \{\mathcal{L}(P, \lambda)\}$.
- Let $\hat{P}(\lambda) = \arg\max_P \{\mathcal{L}(P, \lambda)\}$.
- Let $\lambda^* = \arg\min_\lambda Q(\lambda)$. By LP duality, $\sum_P c_P f_P^*$ = $Q(\lambda^*)$.
- Find λ^* using sequence of "message-passing" updates:

$$\lambda^{m+1} = \lambda^m - \alpha^m A_i\big(\hat{P}(\lambda^m)\big)$$

- Subgradient $A_i\big(\hat{P}(\lambda^m)\big)$ computed w/ Viterbi algorithm.
- Appropriate step size α^m assures convergence to λ^*.
- May take a long time to converge to LP optimum.

Tree-Reweighted Max-Product

- General MAP estimation algorithm [Wainwright, Jaakkola, Willsky, Allerton '02].
- On turbo-like codes: simple message passing decoder.
- Same "cost adjustments" λ as subgradient decoding.
- Messages computed using log-likelihood ratio (LLR):

$$\lambda_i^{m+1} = \lambda_i^m + \alpha^m \big(\mathrm{LLR}(\lambda^m; \hat{t}) - \mathrm{LLR}(\lambda^m; t) \big)$$

- If sign($\mathrm{LLR}(\lambda; \hat{t})$) = sign($\mathrm{LLR}(\lambda; t)$), for all $X_i = \{t, \hat{t}\}$:
  $\rightarrow$ the constituent codes (repeater, accumulator) agree on a codeword.

Tree-Reweighted Max-Product

- General MAP estimation algorithm [Wainwright, Jaakkola, Willsky, Allerton '02].
- On turbo-like codes: simple message passing decoder.
- Same "cost adjustments" λ as subgradient decoding.
- Messages computed using log-likelihood ratio (LLR):

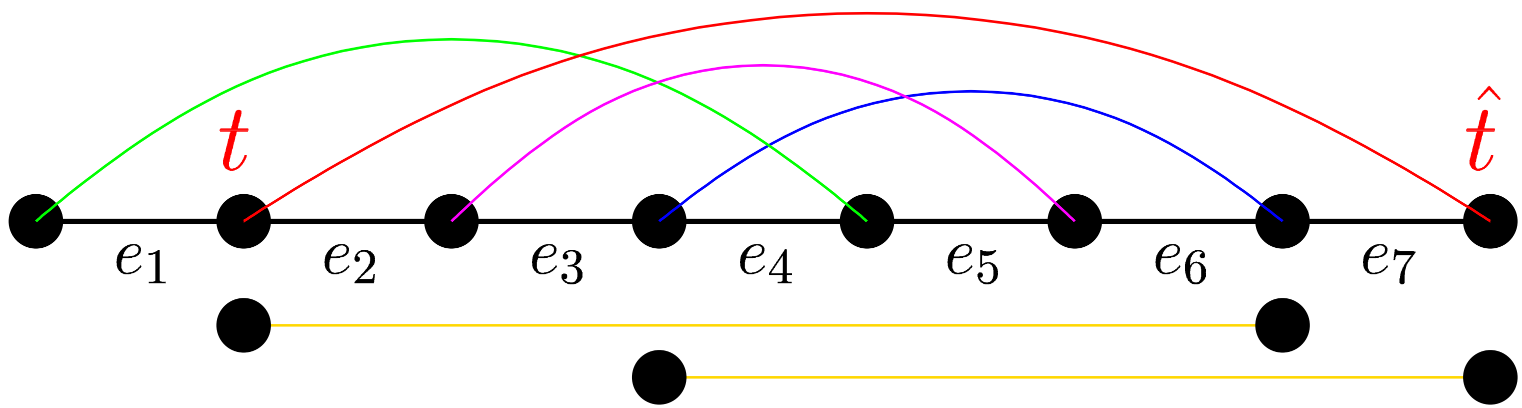
$$\lambda_i^{m+1} = \lambda_i^m + \alpha^m \big(\mathrm{LLR}(\lambda^m; \hat{t}) - \mathrm{LLR}(\lambda^m; t) \big)$$

- If sign($\mathrm{LLR}(\lambda; \hat{t})$) = sign($\mathrm{LLR}(\lambda; t)$), for all $X_i = \{t, \hat{t}\}$:
  $\rightarrow$ the constituent codes (repeater, accumulator) agree on a codeword.

- By LP duality,
  $\rightarrow$ TRMP has found the ML code word.

Promenades

- Precise characterization of noise patterns that cause decoding error for BSC, AWGN.
- Let G be a particular weighted, undirected graph:



$$\text{cost}(e_j) = \begin{cases} -1 & \text{if bit j flipped by channel} \\ +1 & \text{otherwise} \end{cases}$$

- A *promenade* is a collection D of subpaths of G, where
  - For all $X_i = \{t, \hat{t}\}$, $deg_t(D) = deg_{\hat{t}}(D)$.
- $deg_t(D)$ = number of subpaths in D that start or end at t.

Noisy Promenades

- The *cost* of a promenade is the sum of the costs of its subpaths.
  - A *noisy* promenade is one whose cost is less than or equal to zero.

Theorem [FeKa02]: RALP makes a decoding error iff G has a noisy promenade.

- Natural generalization to AWGN, any turbo-like code.

Noisy Promenades

- The *cost* of a promenade is the sum of the costs of its subpaths.
  - A *noisy* promenade is one whose cost is less than or equal to zero.

  Theorem [FeKa02]: RALP makes a decoding error iff G has a noisy promenade.

- Natural generalization to AWGN, any turbo-like code.
- Rate-$1/R$ RA codes, $R \geq 3$:
  - Combinatorics tricky (future work).

Noisy Promenades

- The *cost* of a promenade is the sum of the costs of its subpaths.
  - A *noisy* promenade is one whose cost is less than or equal to zero.

Theorem [FeKa02]: RALP makes a decoding error iff G has a noisy promenade.

- Natural generalization to AWGN, any turbo-like code.
- Rate-$1/R$ RA codes, $R \geq 3$:
  - Combinatorics tricky (future work).
- Rate-1/2 RA codes:

Theorem [FeKa02]: Pr[noisy promenade] $\leq n^{-\epsilon}$, if:

$$p \leq 2^{-4(\epsilon+(\log 24)/2)} \text{ (BSC)} \qquad \sigma^2 \leq \frac{\log e}{4+2\log 3+4\epsilon} \text{ (AWGN)}$$

Conclusions

- LP decoding of turbo-like codes:
  - Precise characterization of noise patterns that cause decoding error for BSC, AWGN: “noisy promenades.”
  - Rate-1/2 RA codes: WER $\leq n^{-\epsilon}$.
  - ML certificate property.

- New iterative algorithms for decoding turbo-like codes:
  - Subgradient decoding: converges to LP solution.
  - TRMP: finds ML code word when LLRs agree.

Open Questions

- Better WER bound for rate-$1/R$ RA using noisy promenades ?
  - Conjecture: LP decoding WER $\leq e^{-(cn^{\epsilon})}$.
- WER bounds for other turbo-like codes?
- (Poly-time) convergence proof for TRMP?
- Relationship to sum- and max-product?