Improved bounds on the word error probability of RA(2) codes with linear programming based decoding

Nissim Halabi
Tel-Aviv University
Joint work with Guy Even
Outline

• “Turbo-like” codes.
  – classic Turbo codes & “turbo-like” codes.
  – Repeat Accumulate codes.

• Auxiliary graphs and promenades.

• RALP decoding.

• Characterization of RALP failure.
  – Non-positive cost minimal promenades.
  – Skeleton graphs and skeleton promenades.

• Algorithms for error bounds.

• Experimental results.
Turbo Codes

[Berrou, Glavieux, Thitimajashima, 1993]:

Repeat Accumulate Codes $RA(q)$
[Divsalar, Jin, McEliece, 1998]

$RA(2)$ code:

Information word:

Repeat:

Interleave:

Accumulate (codeword): 1 1 0 1 1 0 0 0 0
**Auxiliary Graphs of RA(2) codes**

- **Model for RA(2) codes** [Bazzi *et al.* 2001].
- **Undirected graph**: path + matching.
  - Vertices: codeword bits
  - Matching edges: interleaver;
  - Path edges: also called *Hamiltonian* edges.

- **Theorem** [BMMS01]: code distance = graph’s girth (shortest cycle).
- **Construct maximal distance RA(2) codes**: cubic graphs with girth $\Theta(\log n)$ [Erdös & Sachs, 1963][BMMS01].
Auxiliary Graphs of RA(2) codes (cont.)

- Error word $\mapsto$ costs to Hamiltonian edges [Feldman & Karger, 2002].
  - BSC:
    $$ c[e_i] = \begin{cases} 
    -1 & \text{if bit } i \text{ is flipped by the channel} \\
    +1 & \text{otherwise} 
    \end{cases} $$
  - AWGN channel:
    $$ c[e_i] = 1 + \varphi_i \quad \text{where } \varphi_i \sim \mathcal{N}(0, \frac{N_0}{2}) $$

- Matching edge: cost $\equiv 0$. 
Promenade [FK]

- A *Promenade* is a closed walk that does not traverse an edge twice in a row.
- The *cost* of a promenade is the sum of the costs of the edges traversed by the promenade.
- Infinitely many promenades.
- At least every second edge is Hamiltonian.

\[ c[M] = -1 \]
Promenade [FK]

- A *Promenade* is a closed walk that does not traverse an edge twice in a row.
- The *cost* of a promenade is the sum of the costs of the edges traversed by the promenade.
- Infinitely many promenades.
- At least every second edge is Hamiltonian.

\[ c[M] = +3 \]
Promenade [FK]

- A Promenade is a closed walk that does not traverse an edge twice in a row.
- The cost of a promenade is the sum of the costs of the edges traversed by the promenade.
- Infinitely many promenades.
- At least every second edge is Hamiltonian.

\[ c[M] = 0 \]
RALP decoding [FK]

• A provably polynomial time algorithm.
• Agrees with ML decoding; or outputs “error”.
• Theorem [FK02]: The RALP decoder succeeds if all promenades have positive cost. The RALP decoder fails if there is a promenade with negative cost.
  – Success: output the original information word.
• \( \mathbb{P}(\text{fail}) \leq \mathbb{P}\{\exists \text{ promenade } M : c[M] \leq 0\} \)
• Theorem [FK02]: \( \mathbb{P}(\text{fail}) \leq \frac{1}{\text{poly}(n)} \)
  – Specific, deterministically constructible codes.
  – Every code length.
Our Results

• New structural theorem that characterizes the event that RALP fails.
• Present polynomial time algorithms that, given an RA(2) code, compute upper and lower bounds on $P_w$.
• Experiments demonstrate an improvement for bounds on $P_w$. 
NPCM-Promenades

• Non-Positive Cost Minimal Promenade:
  A promenade with:
  – Non-positive cost
  – Minimal with respect to inclusion.

• Observation: ∃ NPCM-promenade ⇔ ∃ non-positive cost promenade.

• The number of NPCM-promenades is finite.
Skeleton Graphs and Promenades

• A skeleton graph has the structure of a “tree of cycles”.

![Diagram](image-url)
Skeleton Graphs and Promenades

- A skeleton graph has the structure of a “tree of cycles”.
- A skeleton promenade is a closed Eulerian tour induced by a tree of cycles.
Skeleton Graphs and Promenades

- A skeleton graph has the structure of a "tree of cycles".
- A skeleton promenade is a closed Eulerian tour induced by a tree of cycles.
- A skeleton walk is a sub-walk of a skeleton promenade.
Characterization of RALP-failure

Theorem: Every NPCM-promenade is a skeleton promenade.

- characterization $\rightarrow$ bound

\[ \Pr\{\text{fail}\} \leq \Pr\{\exists \text{skeleton promenade } M \land c[M] \leq 0\} \]

- $g \triangleq \text{girth} \ (= \log n)$

- Distinction between two types of promenades:
  - Short promenades: length $< 2g + 2$ $\rightarrow$ $P_{short}$
  - Long promenades: length $\geq 2g + 2$ $\rightarrow$ $P_{long}$
Short and Long promenades - Intuition

\[ \Pr\{c[e_i] = -1\} = p \ll 1 \]

- Promenade is simple \(\Rightarrow\)
  \[ E\{c[\text{prom.}]\} \geq \frac{1}{2} \cdot |\text{prom.|(1 - 2p)} \gg 0 \]

- Chernoff bounds \(\Rightarrow\)
  - Long promenades are “easy”
  - Short promenades are “hard”

Problem: repetitions (dependency).
**Short NPCM-Promenades (length < 2g+2)**

- Few Hamiltonian edges; Few errors $\Rightarrow$ non-positive cost
- Claim: every short NPCM-promenade is a simple cycle, namely:

  \[ P_{\text{short}} = \Pr\{\exists \text{ simple short cycle } C : c[C] \leq 0\} \]

- For a cycle $C$ with $h$ Hamiltonian edges:

  \[ \Pr\{c[C] \leq 0\} = \sum_{i=\left\lceil \frac{h}{2} \right\rceil}^{h} \binom{h}{i} p^i (1 - p)^{h-i} \]

  Majority of Hamiltonian edges are negative.
Short NPCM-Promenades (Cont.)

- One can enumerate all short simple cycles in polynomial time.

- Lower bound:
  - Consider cycles with fewest Hamiltonian edges.
  - Deal with intersections of cycles: Compute \( \Pr \{ \exists \text{ cycle } C : c[C] \leq 0 \} \) using inclusion-exclusion principle.
Long NPCM-Promenades ($\text{length} \geq 2g+2$)

- Lemma: If there exists a long NPCM-promenade $M$, then there exists a non-positive cost skeleton walk that contains $g + 1$ Hamiltonian edges (with repetitions).

$$P_{\text{long}} \leq \Pr\{\exists \text{skeleton walk } W : c[W] \leq 0 \& \text{ham}(W) = g + 1\}$$

- Computed similarly to the tree-bound of Feldman et al. [FKW02].
## Experimental Results

\( n = 1024, \ g = 10 \); values in log scale (\( \log_{10} \))

<table>
<thead>
<tr>
<th>( p )</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>skeleton-bound</td>
<td>-1.42</td>
<td>-6.39</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_w ) Lower Bound</td>
<td>-3.53</td>
<td>-6.52</td>
<td>-9.52</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_{long} ) Upper Bound</td>
<td>-1.43</td>
<td>-7.19</td>
<td>-12.49</td>
<td>-17.53</td>
</tr>
<tr>
<td>( P_{short} ) Upper Bound</td>
<td>-3.13</td>
<td>-6.47</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>Tree-bound [FKW02]</td>
<td>No Bound</td>
<td>-2.75</td>
<td>-5.75</td>
<td>-8.75</td>
</tr>
</tbody>
</table>
### Experimental Results

\( n = 1024, \ g = 10 \) ; values in log scale \((\log_{10})\)

<table>
<thead>
<tr>
<th>p</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>skeleton-bound</td>
<td>-1.42</td>
<td>-6.39</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_w ) Lower Bound</td>
<td>-3.53</td>
<td>-6.52</td>
<td>-9.52</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_{long} ) Upper Bound</td>
<td>-1.43</td>
<td>-7.19</td>
<td>-12.49</td>
<td>-17.53</td>
</tr>
<tr>
<td>( P_{short} ) Upper Bound</td>
<td>-3.13</td>
<td>-6.47</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>Tree-bound [FKW02]</td>
<td>No Bound</td>
<td>-2.75</td>
<td>-5.75</td>
<td>-8.75</td>
</tr>
</tbody>
</table>

- NPCM-promenades characterization → Improve previous bounds by \( \sim \times 1000 \).
Experimental Results

\( n = 1024, \ g = 10 \); values in log scale (\( \log_{10} \))

<table>
<thead>
<tr>
<th>( p )</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>skeleton-bound</td>
<td>-1.42</td>
<td>-6.39</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_w ) Lower Bound</td>
<td>-3.53</td>
<td>-6.52</td>
<td>-9.52</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_{long} ) Upper Bound</td>
<td>-1.43</td>
<td>-7.19</td>
<td>-12.49</td>
<td>-17.53</td>
</tr>
<tr>
<td>( P_{short} ) Upper Bound</td>
<td>-3.13</td>
<td>-6.47</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>Tree-bound [FKW02]</td>
<td>No Bound</td>
<td>-2.75</td>
<td>-5.75</td>
<td>-8.75</td>
</tr>
</tbody>
</table>

- Upper and lower bounds are close (\( p \to 0 \)).
## Experimental Results

\( n = 1024, \quad g = 10 \); values in log scale (log\(_{10}\))

<table>
<thead>
<tr>
<th>( p )</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>skeleton-bound</td>
<td>-1.42</td>
<td>-6.39</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_w ) Lower Bound</td>
<td>-3.53</td>
<td>-6.52</td>
<td>-9.52</td>
<td>-12.52</td>
</tr>
<tr>
<td>( P_{long} ) Upper Bound</td>
<td>-1.43</td>
<td>-7.19</td>
<td>-12.49</td>
<td>-17.53</td>
</tr>
<tr>
<td>( P_{short} ) Upper Bound</td>
<td>-3.13</td>
<td>-6.47</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>Tree-bound [FKW02]</td>
<td>No Bound</td>
<td>-2.75</td>
<td>-5.75</td>
<td>-8.75</td>
</tr>
</tbody>
</table>

- Short promenades determine \( P_w \). \( (P_{short} \xrightarrow{p \to 0} P_w) \)
Experimental Results

RA(2) LP Based Decoding, n = 1024, g = 10

Diagram showing the relationship between Shannon Capacity and BSC Crossover Probability with a line labeled 'Tree Bound'.
Experimental Results

RA(2) LP Based Decoding, \( n = 1024, \ g = 10 \)

- \( P_w \) (Probability of Word Error)
- Shannon Capacity
- Lower Bound
- Tree Bound
Experimental Results

RA(2) LP Based Decoding, $n = 1024$, $g = 10$
Experimental Results

RA(2) LP Based Decoding, $n = 1024$, $g = 10$
Experimental Results

RA(2) LP Based Decoding, n = 1024, g = 10

- Probability of Word Error vs. BSC Crossover Probability
- Shannon Capacity
- Upper Bound
- Lower Bound
- Skeleton Bound
- Tree Bound
- $P_{short}$
- $P_{w}$
- $P_{long}$

Graph shows the comparison of different bounds and probabilities for a given decoding method with parameters n = 1024 and g = 10.
Conclusion

• New characterization of RALP decoding failure.
• Efficient algorithms for computing upper- & lower-bounds on $P_w$.
• Experimental results:
  – $P_w$ smaller by $\sim \times 1000$.
  – Lower bound close to upper bound.
Open problems

• Bound for specific RA(3) codes.
• Coding theorem for RA(3).
Experimental Results

RA(2) LP Based Decoding, $n = 1024$, $g = 10$
## Experimental Results

\( n = 1024, \ g = 10 \); values in log scale (\(\log_{10}\))

<table>
<thead>
<tr>
<th>(p)</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>skeleton-bound</td>
<td>-1.42</td>
<td>-6.39</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>(P_w) Lower Bound</td>
<td>-3.53</td>
<td>-6.52</td>
<td>-9.52</td>
<td>-12.52</td>
</tr>
<tr>
<td>(P_{long}) Upper Bound</td>
<td>-1.43</td>
<td>-7.19</td>
<td>-12.49</td>
<td>-17.53</td>
</tr>
<tr>
<td>(P_{short}) Upper Bound</td>
<td>-3.13</td>
<td>-6.47</td>
<td>-9.51</td>
<td>-12.52</td>
</tr>
<tr>
<td>Tree-bound [FKW02]</td>
<td>No Bound</td>
<td>(\boxed{-2.75})</td>
<td>-5.75</td>
<td>-8.75</td>
</tr>
<tr>
<td>exp-tree-bound</td>
<td>No Bound</td>
<td>(\boxed{-2.93})</td>
<td>-5.93</td>
<td>-8.93</td>
</tr>
</tbody>
</table>

- Applying universal bounds to specific RA(2) codes → Minor improvement.
Experimental Results

RA(2) LP Based Decoding, $n = 1024$, $g = 10$