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

(EXTENDED ABSTRACT)

Guy Even
Dept. of Electrical Engineering
Tel-Aviv University
Tel-Aviv 69978, Israel
guy@eng.tau.ac.il

Nissim Halabi
Dept. of Electrical Engineering
Tel-Aviv University
Tel-Aviv 69978, Israel
nissimh@eng.tau.ac.il

Abstract

This paper deals with the linear programming based decoding algorithm of Feldman and Karger [10] for Repeat-Accumulate "Turbo-like" codes. We present a new structural characterization that captures the event that decoding fails. Based on this structural characterization, we develop polynomial algorithms that, given an $RA(2)$ code, compute upper and lower bounds on the word error probability P_w for the binary symmetric channel and the AWGN channel. Experiments with an implementation of these algorithms for bounding P_w demonstrate in many interesting cases an improvement in the upper bound on the word error probability by a factor of over 1000 compared to the bounds by Feldman *et al.* [10, 11, 9]. The experiments also indicate that this improvement increases as the codeword length increases and the channel noise decreases. We also computed lower bounds on the word error probability in our experiments that are roughly 10 times smaller than the upper bound.

1 Introduction

Turbo codes achieve reliable communication at data rates that are close to the channel capacity and are amenable to practical encoding and decoding [3]. However, there are very few proven theorems that guarantee the empirical results. The first such theorem was proven by Divsalar *et al.* [5]. They considered Maximum Likelihood (ML) decoding of randomly generated Repeat-Accumulate $RA(q)$ "Turbo-like" codes. They proved, for $q > 2$, that if the noise in the channel is under a certain constant threshold, then the word error probability is asymptotically bounded by an inverse polynomial in the code length. However, random generation of codes does not point to any specific code with such performance. Moreover, ML decoding of linear codes is NP-Complete [2].

Recently Feldman and Karger [10] introduced a breakthrough in the research of Turbo codes. They presented a novel decoding algorithm for $RA(q)$ codes (for every $q \geq 2$) called RALP that is based on linear programming. For $RA(2)$ codes, they proved that under the binary symmetric channel (BSC) with a certain constant threshold bound on the noise, the word error probability of RALP is bounded by an inverse polynomial in the code length. A similar claim was also shown for the Additive White Gaussian Noise (AWGN) channel [8]. They improved previous results in two important respects: (i) RALP is a provably polynomial time decoding algorithm; when successful RALP provides a certificate that it agrees with ML decoding. (ii) They presented a deterministic polynomial algorithm for constructing $RA(2)$ codes, for every codeword length. The constructed codes achieve the bounds on the word error probability, and, in particular, they do not rely on a random interleaver.

We briefly outline the analysis of the RALP decoding algorithm for $RA(2)$ in [10] (more details appear in the next section). An auxiliary graph is attached to the code. Edges in the auxiliary graph are assigned costs in $\{-1, 0, 1\}$ according to the errors introduced by the BSC[1]. The analysis considers closed walks, called *promenades*. The *cost* of a promenade is the sum of the costs of the edges traversed by the promenade. Feldman and Karger [10] proved that if RALP decoding of an $RA(2)$ code fails, then there exists a non-positive cost promenade in the auxiliary graph. The coding theorem for $RA(2)$ codes is based on bounding the probability that a non-positive cost promenade exists.

Our first contribution is a new structural theorem that characterizes the event that RALP fails. This characterization states that if RALP fails, then there exists a tree-of-cycles subgraph in the auxiliary graph such that every Eulerian tour induced by this tree-of-cycles has non-positive cost. We refer to such Eulerian tours as *skeleton promenades*. Apart from providing a neat characterization of RALP decoding errors, we employ the structural theorem to compute improved bounds on the word error probability (P_w). We develop polynomial time algorithms that, given an $RA(2)$ code, compute upper and lower bounds for P_w.

We outline the techniques we use for computing the bounds for P_w. Instead of applying a union bound on the probability of a family of events that includes the event that RALP fails, we apply our structural characterization of RALP's failure. We focus on non-positive cost skeleton promenades and distinguish between long and short such promenades. For the long promenades, we bound the probability that RALP fails due to such promenades using a recursive procedure similar to the tree bound of Feldman *et al.* [11, 9]. Our structural theorem implies that the short promenades must be simple cycles. Hence we enumerate all the short cycles. For cycles whose length is far from the girth, we apply the union bound. However, for cycles whose length (nearly) equals the girth, we apply bounds based on the inclusion-exclusion principle. This approach enables us to compute far better upper bounds on P_w as well as a lower bound.

We implemented our algorithm for bounding the word error probability of $RA(2)$ codes using RALP decoding. Our experiments show an improvement compared to the bounds of [10, 11, 9] that increases as (i) the codeword length increases, and (ii) the crossover probability decreases. In many interesting cases, the improvement is by a factor of over 1000. Moreover, our experiments indicate a small gap between the upper and lower bounds. This limits further improvements in bounding the word error probability of $RA(2)$ codes using RALP decoding. We point out that our algorithm works also for very small values of P_w; such small values cannot be estimated using standard simulations.

We present the algorithms and the experimental results with respect to the binary symmetric channel. We deal with the AWGN channel in the full version [7]. Due to space limitations, proofs as well as other details are omitted; more details can be found in the full version.

[1] In an AWGN channel the edge costs are Gaussian random variables.

2 Preliminaries

Graph terminology. Let $G = (V, E)$ denote an undirected graph with edge costs $c[e]$. A *walk* W (of length k) in a graph G is a non-empty alternating sequence

$$W = \langle v_0, e_0, v_1, e_1, \ldots, v_{k-1}, e_{k-1}, v_k \rangle$$

of vertices and edges in G such that $e_i = (v_i, v_{i+1})$ for all $i < k$. If $v_0 = v_k$, the walk is *closed*. A *path* is a walk with no repeated vertex. A *cycle* is a closed walk where the only repeated vertex is the first and last vertex. The cost $c[W]$ of a walk W is defined by $c[W] = \sum_{i=0}^{k-1} c[e_i]$.

Given a walk W, we denote by $E(W)$ the multi-set consisting of undirected edges in W. Namely:

$$E(W) = \{e_i \ : \ e_i \in W\}.$$

Note that two instances of the same edge in $E(W)$ may be traversed in opposite directions by W. Let A and B be two multi-sets. Inclusion of multisets, $A \subseteq B$, means that, for every element $x \in A$, the multiplicity of x in A is not greater than the multiplicity x in B.

Definition 1 ([10]). *A* promenade *M in a graph G is a closed walk such that the same edge does not appear twice in a row, i.e., $e_i \neq e_{i+1 \ (mod \ |W|)}$, for every i.*

RA(2) codes. Repeat-accumulate codes were defined by Divsalar *et al.* [5] as examples of "turbo-like" codes with proven word error probability for random interleavers with respect to maximum likelihood decoding. Following Feldman and Karger [10], we focus on $RA(2)$ codes. In an $RA(2)$ code the block size is $n = 2k$, where k is the information word length. An $RA(2)$ encoder uses the following steps: (i) repeat every information bit twice, (ii) send the n-bit string through an *interleaver* that permutes the bits, and finally (iii) send the permuted string through an *accumulator* that outputs the parity of the substring seen so far.

RALP decoding. Feldman and Karger [10] attach an *auxiliary graph* $G = (V, E)$ to an $RA(2)$ code $\mathcal{C}$ as follows. The vertex set V is simply $\{v_1, \ldots, v_n\}$, where v_i represents the ith bit in a codeword. The edges are of two kinds: (i) *Hamiltonian* edges (v_i, v_{i+1}), for every $1 \leq i < n$, and (ii) *matching* edges (v_i, v_j), where i and j are images of the same information bit position with respect to the interleaver (i.e., $\pi^{-1}(j) = \pi^{-1}(i) + 1$ and $\pi^{-1}(j)$ is even, where $\pi : [1, n] \rightarrow [1, n]$ denotes the permutation implemented by the interleaver).

In a BSC the edges of the auxiliary graph are assigned cost $c[e] \in \{-1, 0, 1\}$ as follows: Matching edges always have zero cost. The Hamiltonian edge (v_i, v_{i+1}) is assigned cost 1 if v_i is not flipped by the channel; otherwise it is assigned cost (-1).

Feldman and Karger introduced a linear programming based decoding algorithm called RALP and proved the following theorem.

Theorem 2 ([10]). *The RALP decoder for $RA(2)$ codes succeeds if all promenades in G have positive cost. The RALP decoder fails if there is a promenade in G with negative cost.*

When the RALP decoder *succeeds* it returns the original information word. Moreover, whenever the RALP decoder succeeds, it also provides a proof (called a *certificate*) that the decoded word agrees with ML decoding. Hence, an upper bound on the probability that the RALP decoder fails is also an upper bound on the probability that ML decoding fails.

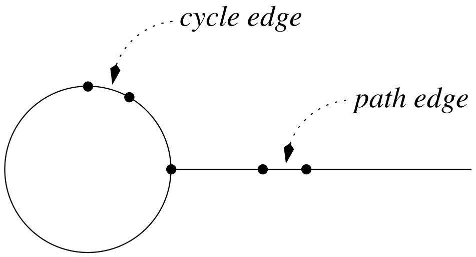

Figure 1: A finger.

3 Non-Positive Cost Minimal Promenades

In this section we introduce a special class of promenades which we call *Non-Positive Cost Minimal Promenades* (NPCM-promenades, in short). We also introduce a class of sub-graphs which we call *skeleton graphs*. The skeleton graphs of G induce a particular set of promenades in G called *skeleton promenades*. Our main structural theorem is a characterization of all non-positive cost minimal promenades as skeleton promenades.

Note that the results described in this section do not rely on any assumptions on the properties of the graph G (e.g. degree, structure etc.) or on the weight function $c : E \to \mathbb{R}$.

NPCM-promenades. We study minimal promenades since if there exists a non-positive cost promenade, then there must exist such a minimal promenade.

Definition 3 (NPCM-promenade). *A promenade M is a* non-positive cost minimal promenade *if: (i) $c[M] \leqslant 0$, and (ii) there is no promenade M' in G which satisfies $E(M') \subsetneqq E(M)$ and $c[M'] \leqslant 0$.*

Lemma 4 (edge multiplicity ≤ 2). *The multiplicity of every edge in an NPCM-promenade is at most two.*

Skeleton graphs. A skeleton graph is a subgraph that has the structure of a "tree of cycles". Figures 1 and 2 are depictions of the following three definitions.

Definition 5 (finger). *A* finger *is a graph obtained by connecting a simple path to a simple cycle.*

We distinguish between two types of edges on a finger: (1) A *path-edge* is an edge which belongs to the path of a finger, and (2) A *cycle-edge* is an edge which belongs to the cycle of a finger.

Definition 6 (skeleton). *A* skeleton *is defined recursively as follows: [Base] A simple cycle is a skeleton. [Recursion step] The following graph is a skeleton: A graph consisting of a skeleton S and a finger F that share a single vertex v s.t. (1) $deg_F(v) = 1$, (2) $deg_S(v) = 2$, and (3) v is a vertex in one of the cycles of skeleton S.*

Definition 7 (skeleton promenade). *A* skeleton promenade *M^S induced by the skeleton S, is a closed Eulerian walk obtained by doubling every path-edge in the skeleton S.*

Theorem 8. *Every non-positive cost minimal promenade in G is a skeleton promenade.*

Theorem 8 has several corollaries; we mention two below.

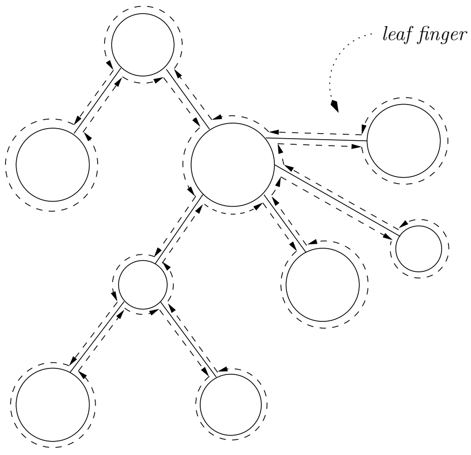

Figure 2: A skeleton graph (tree of cycles) and a skeleton promenade depicted by a dashed line. A skeleton graph G' with ℓ cycles induces 2^{ℓ} different skeleton promenades.

Corollary 9. *If S is an underlying skeleton graph of an NPCM-promenade, then the cost of every cycle in S is positive. Moreover, the cost of a skeleton walk induced by every subtree hanging from one of the cycles in S is negative.*

Corollary 10. *The multiplicity of every vertex in an NPCM-promenade is at most two.*

4 Bounding the word error probability

In this section we present polynomial algorithms that, given an $RA(2)$ code, compute upper and lower bounds on P_w. The correctness of the algorithms is based on the characterization of non-positive cost minimal promenades as skeleton promenades. Our experiments show that further improvement of the upper bound is limited since the upper bound and lower bound for P_w are close.

We describe the algorithm and its results with respect to the BSC. Note that a similar analysis could be applied with a few changes to an additive white Gaussian noise (AWGN) channel [7].

Theorem 2 implies that an upper bound on the probability that G contains a promenade with cost less then or equal to zero, is also an upper bound on the probability that the RALP decoder fails. Our goal is to prove an upper bound on the probability that a non-positive cost promenade exists. Since G is cyclic, there are infinitely many promenades (e.g. one may traverse a cycle an arbitrary number of times). Feldman and Karger [10] suggested to bypass the problem of infinitely many promenades by bounding the number of non-positive cost simple paths containing $\lfloor \frac{g}{2} \rfloor$ Hamiltonian edges. This helps because if a non-positive cost promenade exists, then there exists a non-positive simple path containing $\lfloor \frac{g}{2} \rfloor$ Hamiltonian edges.

We propose a different approach for proving an upper bound. This approach is based on the structure of NPCM-promenades formulated in Theorem 8. Recall that there is an NPCM-promenade in G iff there is a non-positive cost promenade in G.

From Theorem 8 it follows that in order to bound the probability that the RALP decoder fails, it is sufficient to bound the probability that there is a skeleton promenade in G with non-positive cost. Since the multiplicity of every edge and vertex in a skeleton promenade is at most two, it follows that the number of skeleton promenades in a graph G is finite. Unfortunately, the number of skeleton promenades is exponential. We therefore resort to algorithms that bound the number of skeleton promenades in graphs corresponding to specific codes rather than general bounds.

Consider an auxiliary graph G with girth g that corresponds to an $RA(2)$ code $\mathcal{C}$. We consider two cases:

1. There is an NPCM-promenade M in G such that $|M| < 2g + 2$.
2. There is an NPCM-promenade M in G such that $|M| \geqslant 2g + 2$.

We bound separately the probability that each of these two cases occur. The sum of these probabilities is an upper bound on P_w of the $RA(2)$ code $\mathcal{C}$.

4.1 Short NPCM-Promenades

We would like to prove an upper bound on the probability that an NPCM-promenade M of length at most $2g+1$ exists in G. We denote this probability by P_{short}. We refer to a promenade as *short* if it contains at most $2g + 1$ edges.

Short promenades have few Hamiltonian edges, and therefore a few channel transitions can render a short promenade non-positive. Hence, short promenades have a significant influence on P_w. For this reason, rough estimates on P_{short} do not suffice.

Claim 11. *P_{short} equals the probability that there exists a simple short cycle C in G with non-positive cost.*

Proof. By Theorem 8 every non-positive cost minimal short promenade M in G is a skeleton promenade. If a skeleton promenade is not a simple cycle, then its length is at least $2g + 2$ since it contains at least 2 cycles and a bridge that is traversed on both directions. □

Let $Cycles[h]$ denote the number of short simple cycles in G with h Hamiltonian edges. Consider a simple cycle C with h Hamiltonian edges. Since no two matching edges share an endpoint, at least one of every two consecutive edges in a cycle C is Hamiltonian. Therefore, $Cycles[h] = 0$ for every h, $0 \leqslant h < \lceil \frac{g}{2} \rceil - 1$ and

$$Pr\{c[C] \leqslant 0\} = \sum_{i=\lceil \frac{h}{2} \rceil}^{h} \binom{h}{i} p^i (1-p)^{h-i}. \tag{1}$$

By applying a union bound over all short cycles in G, we conclude that

$$P_{short} \leqslant \sum_{h=\lceil \frac{g}{2} \rceil}^{2g+1} Cycles[h] \cdot Pr\{c[C] \leqslant 0 : ham(C) = h\}. \tag{2}$$

We now tighten this bound. Note that for small crossover probabilities, the probability of a cycle having non-positive cost diminishes as the number of Hamiltonian edges in the cycle increases. Therefore, in order to tighten the bound on P_{short} we distinguish between the following cases:

1. $FewHams$: Short cycles $C \in G$ that satisfy either $ham(C) = \lfloor \frac{g}{2} \rfloor$ or $ham(C) = \lfloor \frac{g}{2} \rfloor + 1$. Note that for graphs with odd girth $Cycles[\lfloor \frac{g}{2} \rfloor] = 0$. Let $P_{short'}$ denote the probability that there exists a cycle $C \in FewHams$ with non-positive cost.
2. $ManyHams$: Short cycles that satisfy $ham(C) \geqslant \lfloor \frac{g}{2} \rfloor + 2$. Let $P_{short''}$ denote the probability that there exists a cycle $C \in ManyHams$ with non-positive cost.

We bound separately $P_{short'}$ and $P_{short''}$. Clearly, $P_{short} \leqslant P_{short'} + P_{short''}$. Note that $P_w \geqslant P_{short} \geqslant P_{short'}$.

A Bound on $P_{short'}$. Cluster the set $FewHams$ as follows: (i) Create an auxiliary graph $\Gamma = (V_\Gamma, E_\Gamma)$ where $V_\Gamma = \{v_C : C \in FewHams\}$ and $E_\Gamma = \{(v_{C_1}, v_{C_2}) : \exists e \in E(C_1) \cap E(C_2)\}$. (ii) Decompose V_Γ into connected components $\{U_i\}$ of Γ. (iii) Define a list of cycles per connected component U_i of Γ. We denote by K_i the list corresponding to U_i. We often refer to K_i as a cluster.

Let $P^K_{short'}$ denote the probability that there exists a cycle with non-positive cost in cluster K. Since no two cycles of different clusters share an edge, the probabilities $\{P^{K_i}_{short'}\}$ are independent. Therefore,

$$P_{short'} = 1 - \prod_{\text{clusters } K} (1 - P^K_{short'}) \tag{3}$$

We compute both upper and lower bounds on $P^K_{short'}$ for every cluster K using Bonferroni inequalities [12] with terms that contain products of at most three terms (i.e., inclusion-exclusion with only the first terms). By substituting the upper and lower bounds for $P^K_{short'}$ in Equation 3, we obtain upper and lower bounds for $P_{short'}$, respectively.

A bound on $P_{short''}$. Cycles that are "abundant" with Hamiltonian edges do not require such a precise estimate as the cycles in $FewHams$. The following two steps are used for computing an upper bound on $P_{short''}$: (i) compute $Cycles[h]$ for every h, $\lfloor \frac{g}{2} \rfloor + 2 \leqslant h < 2g+1$. (ii) apply a union bound (using Equation 1) to compute an upper bound on $P_{short''}$.

Time and space complexity The topic of constructing small cubic graphs with large girth was studied by Erdős and Sachs [6] and Sauer [13] (see also Biggs [4]). These constructions are of cubic graphs with girth $g \simeq \log n$. An efficient $O(n^3)$ algorithm for constructing cubic graphs with girth $g \geqslant \frac{1}{2} \log n$ was presented by Bazzi *et al.* [1]. These cubic graphs consist of a path and a matching, and therefore, induce an $RA(2)$ code. In our experiments we used a modification of Bazzi *et al.* [1] to obtain cubic graphs with girth $g \simeq \log n$. For the purpose of analyzing the complexity, we assume that $g = \log_2 n$.

Matching edges do not share an endpoint, and therefore there are at most $\frac{|V_G| \cdot 2^{2g+1}}{2g} = O(\frac{n^3}{\log n})$ cycles with length at most $2g + 1$. We enumerate all the simple cycles in G with length at most l in $O(n \cdot 2^l)$ time and $O(l)$ space. Therefore, computing $Cycles[h]$ for every h, $\lceil \frac{g}{2} \rceil \leqslant h < 2g + 1$, requires at most polynomial time and logarithmic space in n.

We note that every cluster K of $FewHams$ satisfies $|K| \leqslant |FewHams| = O(n \cdot \frac{3^{g/2}}{2g}) = O(\frac{n^{1+(\log 3)/2}}{\log n})$. In addition, every cycle $C \in K$ satisfies $ham(C) \leqslant \lfloor \frac{g}{2} \rfloor + 1$. It follows that the computation of the tighter bound on P_{short} using Bonferroni inequalities requires at most $O(|K|^3 \cdot 2^{\frac{3}{4}g}) = O(\frac{n^{(15+6\log 3)/4}}{\log^3 n})$ time and $O(|FewHams|) = O(n^{1+\log 3/2})$ space (see [7]).

In our experiments $|FewHams|$ and in particular its clusters are very small and the intersections between cycles in the clusters are even smaller (e.g., when $n = 1024$, we had $|FewHams| = 16$ in which the maximum cluster size was 2 and the maximum number of edges that appeared in two cycles in a cluster equaled 2). Therefore, we could even compute $P_{short'}$ precisely very quickly.

4.2 Long NPCM-Promenades

We would like to prove an upper bound on the probability that a non-positive cost minimal promenade M of length at least $2g + 2$ exists in G. We denote this probability by P_{long}.

In spite of the characterization of all non-positive cost minimal promenades as skeleton promenades, we do not have small bounds on the number of such long skeleton promenades.

We refer to a segment of a skeleton promenade as *skeleton walk*. Instead of bounding P_{long} by enumerating long skeleton promenades, we apply a reduction to the existence of a non-positive cost skeleton walk that traverses $g+1$ Hamiltonian edges in G (with repetitions).
The following lemma follows [10, Lemma 3] and Theorem 8.

Lemma 12. *Consider a graph G with girth $g > 3$. If there exists a non-positive cost minimal promenade M in G where $|M| \geqslant 2g+2$, then there exists a skeleton walk W in G with cost $c[W] \leqslant 0$ that contains $g+1$ Hamiltonian edges (with repetitions).*

Let $P_{long'}$ denote the probability that there exists a skeleton walk W such that $c[W] \leqslant 0$ and $ham(W) = g+1$. Lemma 12 implies that $P_{long} \leqslant P_{long'}$. We now focus on bounding $P_{long'}$.

A simple bound on $P_{long'}$: One could compute all skeleton walks in G with exactly $g+1$ Hamiltonian edges (with repetitions). For each skeleton walk, compute the probability of having a non-positive cost. Apply a union bound over all counted skeleton walks to obtain an upper bound on $P_{long'}$.

This technique is applicable both to the BSC and the AWGN channel. We focus on a tighter bound for the BSC.

A tighter bound on $P_{long'}$ for the case of BSC: We compute a tighter bound based on the *tree bound* of Feldman *et al.* [11, 9].

Let $P_{long'(v)}$ denote the probability that there exists a skeleton walk W in G that satisfies: (i) W starts at vertex v, (ii) $c[W] \leqslant 0$, and (iii) $Ham(W) = g+1$ (with repetitions). We apply a union bound on all vertices in G as starting points for the skeleton walks, to obtain an upper bound on $P_{long'}$, i.e., $P_{long'} \leqslant \sum_{v \in V_G} P_{long'(v)}$.

Consider a skeleton walk W that starts at vertex v. In the first step, we may assume without loss of generality that W starts with an Hamiltonian edge. Since the maximum degree in G is three, a sub-walk can be extended by at most two edges (recall that a skeleton walk may not traverse e^{-1} immediately after e).

This motivates the following definition. Let $SE(d, k, \widetilde{W})$ be the probability that a walk W in G exists, where the walk W satisfies the following conditions:

1. W is an extension of the walk $\widetilde{W}$, i.e., W starts at the last vertex of $\widetilde{W}$,
2. $\widetilde{W} \circ W$ does not traverse an edge twice in the same direction,
3. $\widetilde{W} \circ W$ does not repeat a vertex more than twice,
4. $ham(W) = d$, and
5. W contains at least k occurrences of negative cost Hamiltonian edges.

In [7] we present a polynomial-time logarithmic-space algorithm for computing $SE(g+1, \lceil \frac{g+1}{2} \rceil, \langle v \rangle)$ where $g \simeq \log n$. Notice that $P_{long'(v)} \leqslant SE(g+1, \lceil \frac{g+1}{2} \rceil, \langle v \rangle)$ since every non-positive cost skeleton walk with $g+1$ Hamiltonian edges satisfies the above conditions. Thus, we conclude that

$$P_{long'} \leqslant \sum_{v \in V_G} SE(g+1, \lceil \frac{g+1}{2} \rceil, \langle v \rangle).$$

5 Experimental Results

In this section, we present experimental results of our algorithms for bounding P_w with respect to a BSC. These experiments include constructing specific $RA(2)$ codes.

$n(g)$	128(7)				256(8)			
p	-2	-3	-4	-5	-2	-3	-4	-5
skeleton-bound	-0.51	-3.91	-6.80	-9.15	-1.46	-5.61	-8.83	-11.85
P_w Lower Bound	-3.22	-5.22	-7.22	-9.22	-4.00	-7.00	-10.00	-13.00
P_{long} Upper Bound	-0.52	-3.93	-7.01	-10.02	-1.48	-6.00	-10.11	-14.13
P_{short} Upper Bound	-2.63	-5.12	-7.21	-9.22	-2.75	-5.85	-8.85	-11.86
tree-bound [11]	-0.09	-2.09	-4.09	-6.09	*No Bound*	-1.10	-3.10	-5.10
exp-tree-bound [11]	-0.28	-2.27	-4.27	-6.27	*No Bound*	-1.28	-3.28	-5.28

$n(g)$	512(9)				1024(10)			
p	-2	-3	-4	-5	-2	-3	-4	-5
skeleton-bound	-0.48	-5.09	-8.82	-11.95	-1.42	-6.39	-9.51	-12.52
P_w Lower Bound	-5.00	-8.00	-11.00	-14.00	-3.53	-6.52	-9.52	-12.52
P_{long} Upper Bound	-0.48	-5.16	-9.37	-13.40	-1.43	-7.19	-12.49	-17.53
P_{short} Upper Bound	-2.82	-5.95	-8.96	-11.97	-3.13	-6.47	-9.51	-12.52
tree-bound [11]	*No Bound*	-0.80	-2.80	-4.80	*No Bound*	-2.75	-5.75	-8.75
exp-tree-bound [11]	*No Bound*	-0.98	-2.98	-4.98	*No Bound*	-2.93	-5.93	-8.93

Table 1: Bounds on P_w for selected $RA(2)$ codes. All the quantities are in logarithmic scale ($\log_{10}$).

Good codes have auxiliary graphs with a large girth. The algorithm of Bazzi *et al.* [1] constructs cubic graphs with a girth that equals $\frac{1}{2} \cdot \log n$. We modified this algorithm to construct auxiliary cubic graphs whose girth nearly equals $\log n$. Using this modified algorithm, we constructed auxiliary graphs corresponding to codes with codeword length $n \in \{128, 256, 512, 1024\}$ (see Table 1). For each codeword length we constructed $25 - 50$ different auxiliary graphs and picked the graph with the lowest upper bound on P_w.

Feldman *et al.* [10, 11, 9] presented two upper bounds for P_w called the *path-bound* and the *tree-bound*. The tree-bound (which is computed by running a recursive procedure) outperforms the path-bound (which is an analytic bound). Both bounds depend only on n (and not on the specific graph). We suggest a modification of the tree-bound, called the *exp-tree-bound*; this bound is similar to the tree-bound but applies to a specific code. Our experiments show that the improvement offered by exp-tree-bound compared to the tree-bound is by a factor of approximately $10^{-0.2}$.

In Table 1 and Figure 3 we present our experimental results. We refer to our upper bound of P_w as the *skeleton-bound*. Note that the improvement of the skeleton-bound increases as n increases and p decreases. We compute also a lower bound on P_w which is that lower bound on $P_{short'}$. For the tested graphs with 256 and 512 nodes, the gap between the upper and lower bounds is roughly 10.

Acknowledgments. We thank Jon Feldman for continuously sending us versions of his papers and thesis.

References

[1] L. Bazzi, M. Mahdian, S. Miller, and D. Spielman, “The minimum distance of turbo-like codes,” manuscript, 2001.

[2] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, “On the intractability of certain coding problems,“ *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384-386, May 1978.

[3] C. Berrou, A. Glavieux, and P. Thitimajashima, “Near Shannon limit error-correction coding : Turbo Codes,” *Proc. IEEE International Conference on Communications (ICC)*, pp. 1064-1070, Geneva, Switzerland, May 1993.

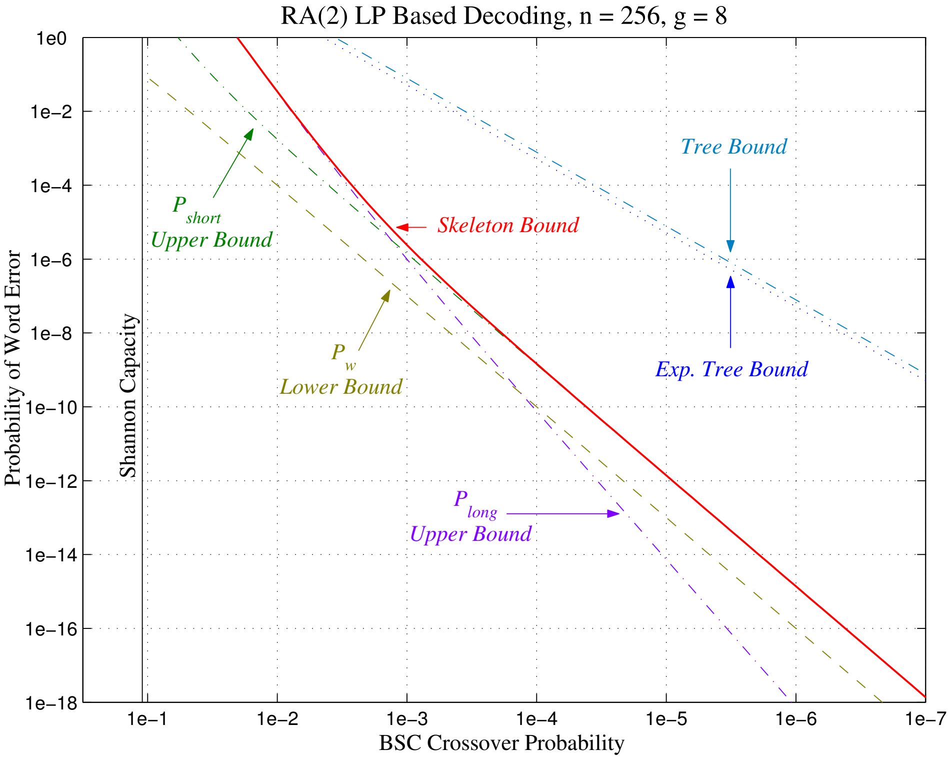

Figure 3: Bounds on P_w for an $RA(2)$ code with corresponding auxiliary graph of size $n = 256$ and girth $g = 8$.

[4] N. Biggs, “Constructions for cubic graphs with large girth,” *Electronic Journal of Combinatorics*, 5(A1), 1998.

[5] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ‘turbo-like’ codes,” *Proc. 36th Allerton Conf. on Communication, Control, and Computing*, pp. 201-210, Sept. 1998.

[6] P. Erdős and H. Sachs, “Reguläre Graphen gegebene Taillenweite mit minimaler Knotenzahl”, Natur.Reine 12, 251–257, 1963.

[7] G. Even and N. Halabi, “Improved bounds on the word error probability of $RA(2)$ codes with linear programming based decoding”, `http://www.eng.tau.ac.il/~guy/Papers/`, 2003.

[8] J. Feldman, “Decoding Error-Correcting Codes via Linear Programming,” Ph.D. Thesis, Massachusetts Institute of Technology, September 2003.

[9] J. Feldman, personal communication, 2003.

[10] J. Feldman and D. R. Karger, “Decoding turbo-like codes via linear programming,” *FOCS*, November 2002.

[11] J. Feldman, D. R. Karger, and M. J. Wainwright, “Linear programming-based decoding of turbo-like codes and its relation to iterative approaches,” *Allerton*, October 2002.

[12] S. Jukna, *Extremal combinatorics : with applications in computer science*, Springer, Berlin, 2001.

[13] N. Sauer, “Extremaleigenschaften regulärer Graphen gegebener Taillenweite, I & II.” Sitzungsberichte Österreich. Acad. Wiss. Math. Natur. Kl., S-B II, 176, 9-25, 27-43, 1967.