6.962: Week 5 Summary of Topic

Presenter:	Stark Draper
Date:	11 October 2000
Paper:	"Irregular Repeat-Accumulate Codes", H. Jin, A. Khandekar, and R. McEliece. *2nd International Symposium on Turbo Codes & Related Topics*, pp. 1–9, Sept, 2000.

1 Summary

In this paper the authors set out to design channel codes that have near-Shannon-limit error correction capabilities, but that also have minimal implementation complexity. Along these lines they pose a "final" problem for channel coding theorists.

For a given channel, find an ensemble of codes that:

1. Has a linear-time encoding algorithm
2. Can be decoded reliably in linear time at rates arbitrarily close to channel capacity.

In this paper the authors introduce a new class of codes called *irregular repeat-accumulate codes* (IRA codes). The performance of these codes can be analyzed, and it is shown that they solve the "final" problem for the binary-erasure channel (BEC). Their performance is also discussed on the additive-white Gaussian noise (AWGN) channel. The code design problem in this case is transformed into a linear programming problem, and simulations are carried out that show very good performance.

2 IRA Codes

Irregular Repeat-Accumulate codes are a generalization of (not surprisingly) Repeat-Accumulate codes [1]. Repeat-Accumulate codes were originally introduced as turbo-like codes that had simple, analyzable structures. What is most surprising about IRA codes is that the complexity of their structures has only been increased marginally (they are still analyzable), yet their performance, vis-a-vis RA codes, has improved greatly. Similar improvements have also been seen in the application of irregular structure to low-density parity-check (LDPC) codes [2], [3]. The structure of IRA codes is shown in Figure 1, and we discuss what we mean by "irregular" below.

In Figure 1 the left column of nodes are the information nodes, the center column are the check nodes, and the right column are the parity nodes. The differences between RA codes and IRA codes are outlined in the following table. The "degree" of a node is the number of edges attached to it. The left (right) degree is the number of edges attached to the left (right) side of the node.

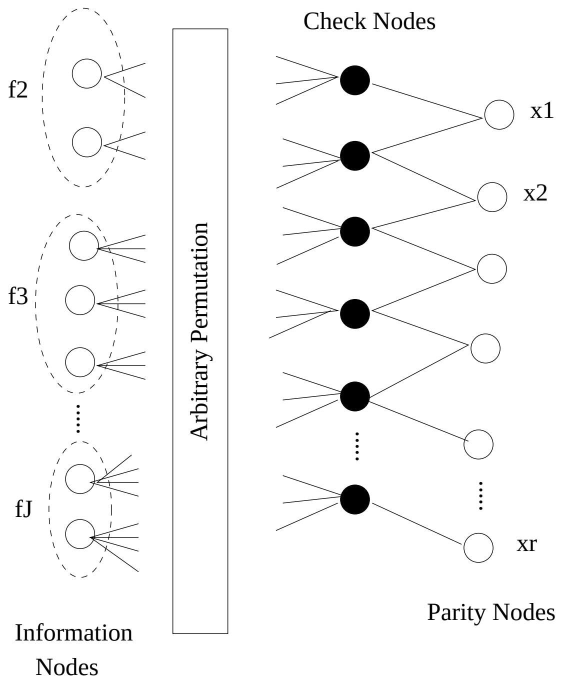

Figure 1: Graphical structure of IRA codes

RA codes	IRA codes
Information nodes of constant degree	Information nodes of variable degree
Check nodes all have left degree = 1	Check nodes all have left degree = a (≥ 1)
Parity nodes each of degree 2	Parity nodes each of degree 2
Nonsystematic (only parity bits sent)	Systematic (info and parity bits sent)

Considering the table above, we see that the 'irregular' structure of the IRA codes comes from the variability of the degree of the info nodes. Information nodes are classified by their degree. The fraction of the information nodes that are of degree K is f_K. In RA codes all the information nodes are of the same degree, e.g. q. Thus there is just one class of nodes f_q, and $f_q = 1$.

We consider the operation of the iterative-decoder for an IRA code. The decoder uses the same sum-product algorithm as discussed by Albert last week. The outgoing message on an edge is a function of the incoming messages on all edges **except** for the outgoing message edge. We subdivide the decoding algorithm into a (slightly artificial, since the steps naturally occur in parallel) cycle of four steps:

0. Information nodes pass messages to check nodes.

1. At a check node, if **no** incoming messages (from any information node or the parity node) is an erasure, the outgoing message (to the other parity node) is the binary sum of all incoming messages. If **any** incoming message is an erasure, the outgoing message is an erasure. (This calculation must be done twice, once for each of the two parity nodes connected to each check node.) Assuming the decoder converges, the steady-state

probability that the outgoing message is an erasure is x_1.

2. At a parity node, the outgoing message (to a check node) is the incoming message if the incoming message is **not** an erasure **or** if the original parity bit was not erased. (This calculation must be done twice, once for each of the two incoming messages.) The steady-state probability that an outgoing message is an erasure is x_2.

3. At a check node, if **no** incoming messages (from each parity node and from all the other information nodes) is an erasure, the output message (to the last info node) is the binary sum of all incoming messages. If **any** incoming message is an erasure, the outgoing message from that node is an erasure. (This calculation must be done a times, once for each of the information nodes connected to each check node.) The steady-state probability that an outgoing message is an erasure is x_3.

4. At an information node, the output message (to each check node) is the input message if any of the incoming messages are **not** erasures **or** if the original parity bit was not erased. (This calculation must be done i times for an information node of degree i, once for each check node to which it is connected.) The steady-state probability that an outgoing message is an erasure is x_0.

As described by Albert last week, the sum-product algorithm only sends a message out from a node along an edge when it receives data (non-erasures) from all other incoming edges. The messages in the case of the BEC (as we will discuss further in class) are the log-likelihood rations, $m = \log(p(0)/p(1)$. If there is an erasure, $p(0) = p(1)$ and $m = 0$. If there is no erasure, there are two possibilities: 1) $p(0) = 1$, $p(1) = 0$, and $m = \infty$ or 2) $p(0) = 0$, $p(1) = 1$, and $m = -\infty$. With these messages the sum-product algorithm simplifies to the cycle of steps outlined above.

Given this decoding algorithm, to show $Pr(erasure) \to 0$ we need one additional fact. This fact is that as the length of a IRA code goes to infinity, the probability that a cycle of any finite length exists in the code goes to zero. This fact was proved for irregular LDPC codes in [2], and the proof is similar for IRA codes. This means that all incoming messages to a node are independent. The proof then follows a number of steps, which we outline for the BEC.

1. Assume the iterative-decoding algorithm is at a fixed point (i.e. all the values are self-consistent, so further iterations will not change any of the information at any node).

2. Since the iterative-decoder is at a fixed point, we follow the iterative algorithm through one cycle (as defined above) and calculate the probabilities of erasure.

   (a) x_1.

   (b) x_2.

   (c) x_3.

   (d) x_0.

3. From the four probabilities solve for x_0.

4. Since x_0 is a probability, $x_0 \in [0, 1]$.
5. Show that $x_0 \notin (0, 1]$. This implies $x_0 \to 0$.

The analysis for the AWGN channel follows that for the BEC pretty closely.

References

[1] H. Jin D. Divsalar and R. J. McEliece. Coding theorems for 'turbo-like' codes. In *Proc. 36th Allerton Conf. on Communication, Control and Computing*, pages 201–210, 1998.

[2] T. J. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Info. Theory*, submitted 1998.

[3] T. J. Richardson and R. Urbanke. Design of provably good low-density parity-check codes. *IEEE Trans. Info. Theory*, submitted 1999.