# 5
# Conclusion and Complementary Work

In this work, we have applied the PEG algorithm to the construction of random graphs in the IRA class. Simulation has demonstrated that this method brings significant advantage over less sophisticated random methods, but does not achieve the theoretical decoding thresholds associated (by density evolution) to the distributions that were used.

In spite of the restrictions imposed on the graph by the PEG algorithm, this method is still essentially a random search through a more restricted population, and the results are not guaranteed to yield the best results. The randomness of the outcomes, combined with the lack of fast algorithms to verify graph properties such as minimum distance, actual decoding thresholds and the statistical distribution of cycles, is one downside of the PEG algorithm.

The diminishing returns of our brute force approach suggest that the use of exact combinatorial designs and a deeper investigation on graph theory for straightforward methods of evaluating distance properties of graphs is still needed. More direct and deterministic methods, where these properties could be obtained without need for repetitive testing and exhaustive search, would be very appealing if they could outperform (or yield similar results to) the PEG algorithm.

Other works such as [SJR2005] have explored the use of combinatorial designs with IRA codes providing codes with similar performance to ours. The best performing design presented in [SJR2005], is called w3IRA and its graph includes parity-nodes with degree three (while in typical IRA codes, they all have degree two). We compared a rate $\frac{1}{2}$ w3IRA code to our equivalent PEG-ST code and the original IRA code presented by Hui Jin, Khandekar & McEliece in [JKM2000] in Figure 5.1. None of the two more recent codes (w3IRA and our PEG-ST) outperformed the one in the paper where IRA codes were originally proposed [JKM2000], where the construction methods are not thoroughly discussed. We cannot, from the available data, observe the performance of the code from [JKM2000] at the range of $\frac{E_b}{N_0}$ values where error floors occur.

We used the PEG algorithm with the distributions provided in

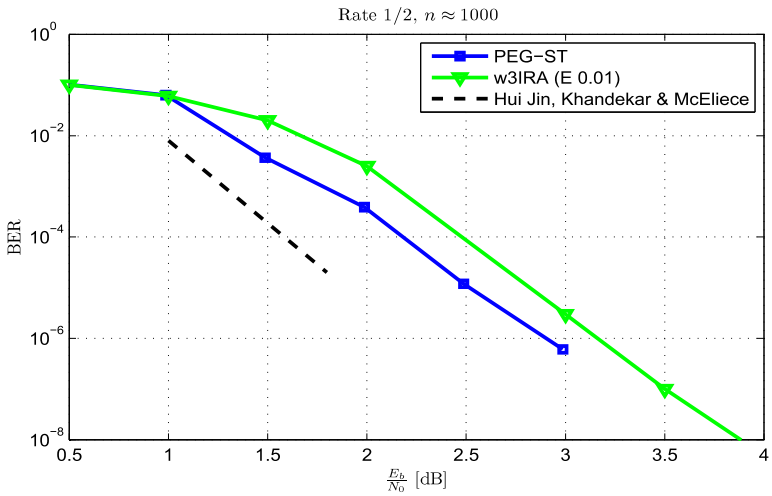


Figure 5.1: Compared performance of methods PEG-ST (distribution #4), w3IRA from [SJR2005] and the code presented in [JKM2000], all with block-length $k = 500$ and rate $R = \frac{1}{2}$.

[JKM2000] and three block lengths that were convenient to our processing capabilities (Intel® Core™ Duo 2.66GHz, 2GB RAM running Matlab® 2006) and available time. The use of the PEG algorithm with IRA codes proved to be effective when used with a suitable degree distribution for the intended block-length, since some degree distributions may not perform well for block-lengths of sizes comparable to the its highest degree. In some cases, distributions with lower degrees and higher decoding $\frac{E_b}{N_0}$ thresholds did outperform distributions with higher degrees and better theoretical decoding thresholds. These peculiarities involving degree distributions motivate a more thorough study of degree distributions and density evolution that was not present in this work.

One investigation that was not made in this work, but could bring further insight on the matter of graph construction, is the construction of graphs attempting to keep all cycles near a deterministic value, based on the girth bounds presented in [HEA2005].