

Optimizing the Error Recovery Capabilities of LDPC-Staircase Codes Featuring a Gaussian Elimination Decoding Scheme: Preliminary Results

Mathieu Cunche, Vincent Roca

► **To cite this version:**

Mathieu Cunche, Vincent Roca. Optimizing the Error Recovery Capabilities of LDPC-Staircase Codes Featuring a Gaussian Elimination Decoding Scheme: Preliminary Results. 2008. inria-00271430

HAL Id: inria-00271430

<https://hal.inria.fr/inria-00271430>

Preprint submitted on 9 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimizing the Error Recovery Capabilities of LDPC-Staircase Codes Featuring a Gaussian Elimination Decoding Scheme: Preliminary Results

Mathieu CUNCHE

Vincent ROCA

INRIA Rhône-Alpes, Planète research team, France, {firstname.name}@inria.fr

I. INTRODUCTION

The erasure recovery capabilities of LDPC-Triangle and LDPC-Staircase codes [2] can be greatly improved by means of a Gaussian elimination decoding scheme [1]. Thanks to this decoding, the LDPC-Triangle codes are now very close to an ideal code. The LDPC-Staircase codes are also improved but they are not as close to an ideal code as the LDPC-Triangle codes are. Furthermore, for some code rates, the performances of the LDPC-Staircase codes diverge from their asymptotic behavior.

This document studies the influence of the N1 parameter of the LDPC-Staircase codes, i.e., the target number of '1s' per column, which in turn controls the left degree of source symbols, on the erasure recovery capabilities of the codes when a Gaussian elimination decoding scheme is used.

II. THE N1 PARAMETER

The N1 parameter, which controls the left degree, is a parameter of the algorithm generating the left side of the parity check matrix. This parameter controls the numbers of '1s' that will be inserted in each column of the left side of the matrix during the first step of the algorithm. This algorithm is the following when this N1 parameter equals 3, the default value [2] :

```

/* Initialize the matrix with 3 "1s" per column, homogeneously */
t = 0;
for (j = 0; j < k; j++) { /* for each source symbol column */
  for (h = 0; h < 3; h++) { /* add 3 "1s" */
    /* check that valid available choices remain */
    for (i = t; i < 3*k && matrix_has_entry(u[i], j); i++);
    if (i < 3*k) {
      /* choose one index within the list of possible
       * choices */
      do {
        i = t + pmms_rand(3*k-t);
      } while (matrix_has_entry(u[i], j));
      matrix_insert_entry(u[i], j);

      /* replace with u[t] which has never been chosen */
      u[i] = u[t];
      t++;
    } else {
      /* no choice left, choose one randomly */
      do {
        i = pmms_rand(n-k);
      } while (matrix_has_entry(i, j));
      matrix_insert_entry(i, j);
    }
  }
}

```

After this step, the generation algorithm checks that there are at least two '1s' in each row of the left side of the parity check matrix. If not, one or two additional '1s' are added to the row :

```

/* Add extra bits to avoid rows with less than two "1s".
 * This is needed when the code rate is smaller than 2/5. */
for (i = 0; i < n-k; i++) { /* for each row */
  if (degree_of_row(i) == 0) {

```

```

    j = pmms_rand(k);
    matrix_insert_entry(i, j);
  }
  if (degree_of_row(i) == 1) {
    do {
      j = pmms_rand(k);
    } while (matrix_has_entry(i, j));
    matrix_insert_entry(i, j);
  }
}

```

Because of this second step, the average number of '1' per columns in the left side of the matrix, i.e., the left degree, can be greater than the specified N1 parameter.

III. RESULTS

A. Erasure recovery capabilities

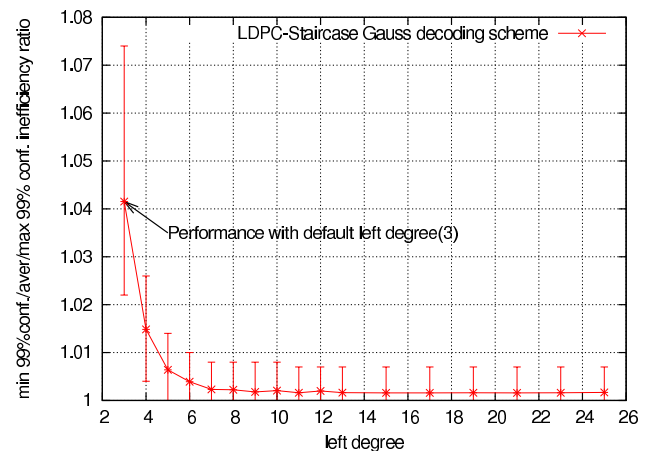


Fig. 1. Inefficiency ratio W.R.T. the N1 parameter (object composed of 1,000 symbols, code rate 2/3)

Figures 1,2,3 show that the inefficiency ratio decreases as the N1 parameter increases.

Figure 4 shows that increasing the N1 value improves the erasure recovery capabilities for the range of code rates considered. More specifically, the choice of the N1 parameter is ideally a function of the code rate, since a value of 4 is often a good choice with a code rate 2/3, while a value of 6 will be preferred with a code rate 1/3.

We can also observe that for lower code rates, the inefficiency ratio no longer depends on the N1 value. This behavior comes from the fact that for low code rates, the specified N1 value is too small to satisfy the condition on the minimum number of '1s' per row. Therefore, after the second step of the algorithm, the actual left degree is always greater than the

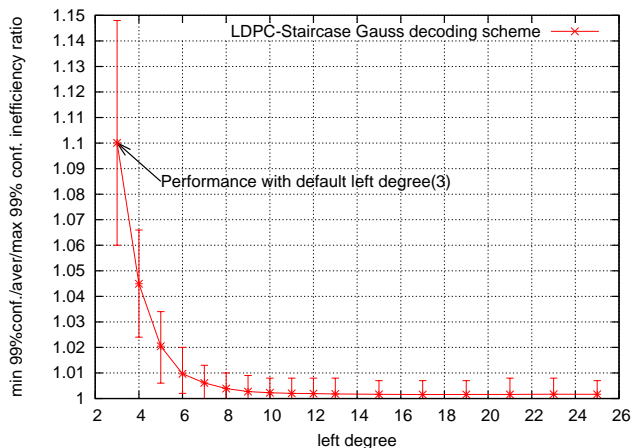


Fig. 2. Inefficiency ratio W.R.T. the N1 parameter (object composed of 1,000 symbols, code rate 2/5)

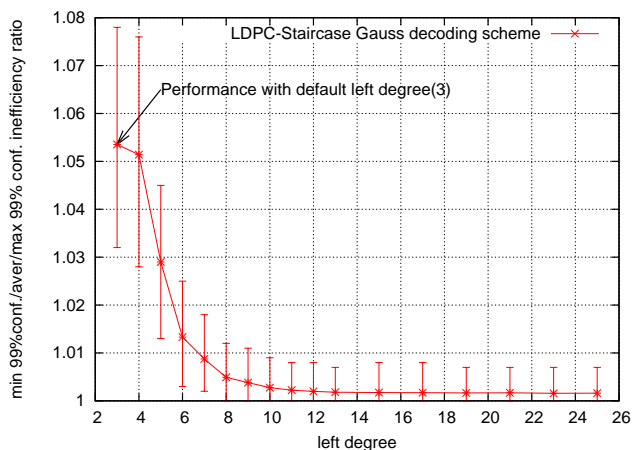


Fig. 3. Inefficiency ratio W.R.T. the N1 parameter (object composed of 1,000 symbols, code rate 1/3)

specified N1 value, and all the curves exhibit similar error correction capabilities.

B. Decoding complexity

The N1 value has an impact on the decoding complexity. Not surprisingly, Figure 5 shows that the decoding time increases with the N1 value. This figure also shows the initial, almost constant, decoding complexity when the Zyablov iterative decoding succeeds (i.e., for low loss probabilities), and later on the progressively increasing complexity of the Gaussian elimination since the iterative decoding is no longer sufficient (i.e., as the loss probability approaches the decoding limit). As N1 increases, on the one hand the Gaussian elimination starts earlier (for lower values of the loss probability), and is more and more costly. Nevertheless, even if the decoding complexity of the LDPC decoder increases as N1 increases, this decoder remains at least 9 times faster than the Reed Solomon decoder for the range of values selected. With this code rate (2/3), choosing a value of 4 for N1 offers a good balance since the iterative decoding remain efficient (iterative decoding remains efficient up to loss probabilities that amount

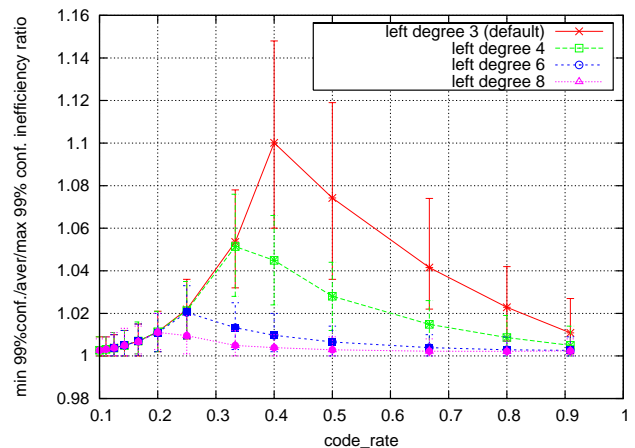


Fig. 4. Inefficiency ratio W.R.T. the code rate (object composed of 1,000 symbols, with various N1 values)

to 27%) while offering a significant improvement in terms of erasure recovery capabilities (the curve ends closer to the theoretical limit of 2/3 than when N1 equals 3).

IV. CONCLUSIONS

The performance of the LDPC-Staircase codes featuring a Gaussian elimination decoding scheme can approach that of an ideal code by a trivial modification of the parity check matrix. Indeed, the erasure correction capabilities are significantly improved when the N1 parameter is increased, where N1 is the target number of '1s' per column in the first step of the parity check matrix creation algorithm. The choice of the N1 parameter is ideally a function of the code rate, since choosing a value of 4 is often a good choice with a code rate 2/3, while a value of 6 will be preferred with a code rate 1/3. A direct consequence is that the N1 value parameter must be transmitted to the LDPC-Staircase decoder as part of the FEC Object Transmission Information.

REFERENCES

- [1] M. Cunche and V. Roca. Improving the decoding of ldpc codes for the packet erasure channel with a hybrid zyablov iterative decoding/gaussian elimination scheme. Research Report 6473, INRIA, March 2008.
- [2] V. Roca, C. Neumann, and D. Furodet. *Low Density Parity Check (LDPC) Forward Error Correction*, January 2008. IETF RMT Working Group, Work in Progress : <draft-ietf-rmt-fec-bb-ldpc-08.txt>.

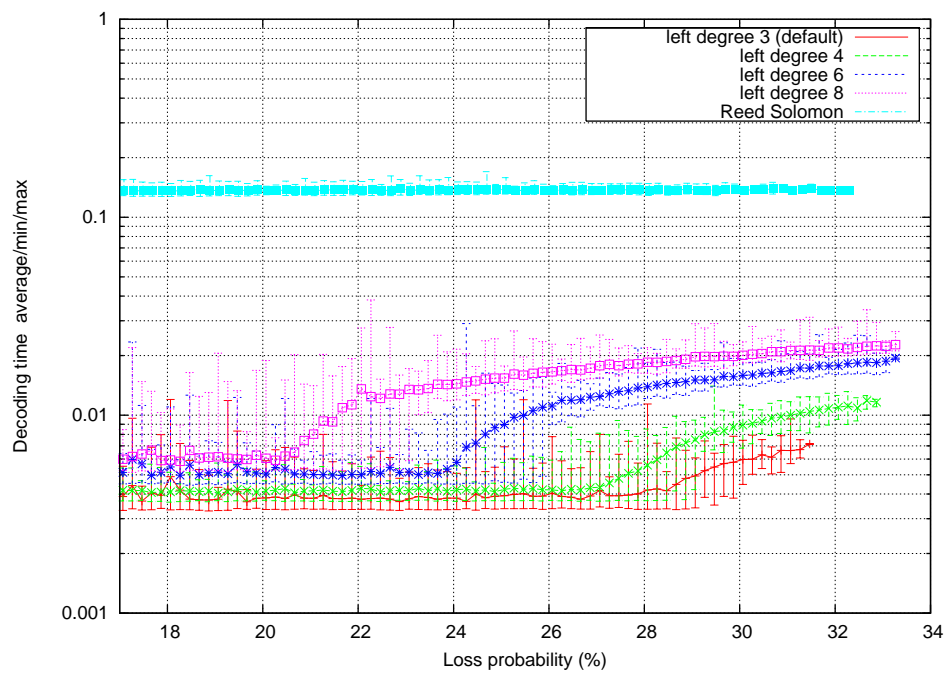


Fig. 5. Decoding time W.R.T. the loss probability with a hybrid Zyablov iterative decoding/Gaussian elimination scheme (object composed of 1,000 symbols, code rate 2/3)