Sparse Degrees Analysis for LT Codes Optimization

Pei-Chuan Tsai

Department of Computer Science
National Chiao Tung University
HsinChu City, TAIWAN
Email: pctsai@nclab.tw

Chih-Ming Chen

Department of Computer Science
National Chiao Tung University
HsinChu City, TAIWAN
Email: ccming@nclab.tw

Ying-ping Chen

Department of Computer Science
National Chiao Tung University
HsinChu City, TAIWAN
Email: ypchen@nclab.tw

Abstract—Luby Transform (LT) codes are a new member in the family of forward error correction codes without a fixed code rate. The property called *rateless* is attractive to researchers in last decade, and lots of studies have been proposed and attempted to improve the performance of LT codes. One variation is the use of a sparse degree distribution instead of a full one referred to in the encoding process of LT codes to reduce the search space. Observing a fact that the ability of a sparse degree distribution is limited by the nonempty degrees, we introduce a tag selection scheme to choose reasonable sparse degrees for LT codes in this paper. We firstly investigate the influence of different degrees on the error rate of LT codes and then propose a general selection algorithm based on our observations. After that, the covariance matrix adaptation evolution strategy (CMA-ES) is applied to find the optimal sparse degree distributions of which the degrees are defined by our selection algorithm. Finally, the experimental results are presented as evidence to show the proposed scheme is effective and practical.

I. INTRODUCTION

Digital fountain [1], [2] is a new class of forward error correction codes, which are used in erasure channel for recovering erased data. The most important property of digital fountain codes is *rateless*. The property allows encoder to generate encoded packets unlimitedly without a fixed code rate. Encoded packets are continuously delivered like a fountain, and any receiver can reconstruct the source data once a sufficient amount of packets are received without considering the order. It means that the performance of digital fountain codes is independent of channel parameters, such as the erasure rate. It is a great advantage to deliver data through an unclear channel or broadcast information to many users whose connection qualities are unequal. The first practical implementation of digital fountain is Luby Transform (LT) codes [3] proposed by Michael Luby in 2002. LT codes are not only rateless but also have low computation complexity due to its simple coding procedure. As other linear block codes, the coding structure of LT codes can be expressed as a tanner graph. Tanner graph is a bipartite graph consists of bit nodes, check nodes and connection edges for indicating the coding relation between the nodes. To generate a check node, LT codes decide the degree of a node according to a particular probability distribution called *degree distribution*. The performance of LT codes totally depends on the size of bit nodes, k , and a given degree distribution. Luby hence suggested two general expressions to define practical degree distributions for LT codes. However, the proposed distribution named as *soliton*

distribution is near optimal only when the size of source data gets close to infinite. In other words, there is still room to improve LT codes with moderate source data size.

The approach to improve the LT codes can be modeled as an optimization problem in which the probability on each degree is considered as a decision variable and the objective is to search for degree distributions with better performance. Many kinds of methods have been proposed to solve such an optimization problem. The researchers firstly focused on optimizing the performance of LT codes for a short data length [4], [5]. [5] even presented an numerical approach which can compute the optimal degree distributions. Unfortunately the optimization scale is limited to $k \leq 30$ due to huge computational cost. For a large k size, [6] made the first attempt to apply heuristic search algorithms to optimize degree distributions. [7] and [8] are our previous studies in which evolutionary algorithms were introduced to address the issue and several distributions better than soliton distributions were obtained for LT codes, while [9] presented the similar idea and focused on maximizing the intermediate recovery rate of LT codes by using multi-objective optimization algorithms.

Studies show the feasibility to optimize degree distributions by evolutionary algorithms. In the optimization framework, the challenge of huge search space is a key issue. The problem dimensionality is equal to the source data length k since an individual should represent the probability on each degree from 1 to k . It is hard to deal with so many variables while k rises to hundreds. Adopting a *sparse degree distribution* is an alternative solution which has been frequently used in LT codes optimization. In a sparse degree distribution, non-zero probabilities distribute on only partial degrees which are predefined by a set of *tags*. The set of tags is a subset of all degrees and used to limit the search in a sub-space whose size is much less than full degrees.

[6] chose the power of 2 to compose sparse degree distributions for optimization, and Fibonacci numbers were used in [7], [8]. All the previous studies proposed practical approaches which can find better distributions in a sparse form. However, the best distribution in sub-space defined by given tags is merely near-optimal for LT codes. Previously, the tags were usually decided according to experimental experience, and there exists no complete investigation on how to choose tags. As a result, this paper employs evolutionary algorithms as a research tool and observes the influence of different

tags on decoding error probability of LT codes. Based on the observations, a sparse degree selection algorithm is proposed to define appropriate tags, and the optimal distribution formed by the selected tags may get close to the global optimal.

The remainder of the paper is organized as follows. Section II gives the details of coding mechanism of LT codes and introduces two evaluation approaches for degree distributions. Section III investigates the probability reallocation of degree distributions and presents the influences of different tags. In section IV, our selection algorithm is proposed and described in detail. Several optimization results with different parameter settings are presented in section V to examine the selection algorithm. Finally, the conclusion and our contribution are given in section VI.

II. LT CODES

Before describing main work of the paper, the operation of LT codes is introduced in this section as a background. The encoding and decoding procedures are given in section II-A. Source data in general are divided into k fragments with an identical length. These fragments are bit nodes or called *input symbols* if the length of each fragment is only a bit. Similarly *encoding symbols* denote the codewords generated by the encoding procedure in LT codes. For a clear presentation, the terms, input symbols, and encoding symbols, are consistently used in this paper. Section II-B introduces *soliton degree distributions*, which were proposed according to a theoretical analysis. Furthermore, because evaluating the quality of degree distributions is necessary in optimization, section II-C introduces two different approaches for evaluating the decoding error rate of LT codes for a given degree distribution.

A. Encoding and Decoding

Given the source data, we suppose that the source data are cut into k input symbols. Before an encoding symbol is generated, a degree d is chosen at random according to the adopted degree distribution $\Omega(d)$, where $1 \leq d \leq k$ and $\sum_{d=1}^k \Omega(d) = 1$. The degree d decides how many distinct input symbols are involved to compose an encoding symbol. Then d input symbols, also named *neighbors*, are chosen uniformly at random and accumulated by XOR. In the design of LT codes, random number plays an essential role in the encoding process. The approach employed by LT codes for a sender to inform receivers of all information is to synchronize a random number generator with a specified seed.

At the receiver side, when n , which is usually slightly larger than k , encoding symbols arrive, *belief propagation* is used to reconstruct the source data step by step. All encoding symbols are initially covered in the beginning. In the first step, all encoding symbols with only one neighbor can be directly released to recover their unique neighbor. For an input symbol that has been recovered but not processed yet, it is called a *ripple* and will be pushed into a queue. At each subsequent step, ripples are popped from the queue as a processing target one by one. A ripple is removed from all encoding symbols that have it as a neighbor. If an encoding symbol has only

one remaining neighbor after removing, the releasing action repeats and may produce new ripples to maintain a stable size of the queue. Maintaining the size of the ripple queue is important because the decoding process fails when the ripple queue becomes empty with uncovered input symbols. In other words, more encoding symbols are required to continue the decoding process. The decoding process succeeds if all input symbols are recovered at the end.

B. Soliton distribution

The coding behavior of LT codes is determined by the degree distribution, $\Omega(d)$, and the number of received encoding symbols, n . Reception overhead, $\varepsilon = n/k$, denotes the delivery efficiency of LT codes, and ε depends on the given degree distribution. Based on the theoretical analysis, Luby proposed the *ideal soliton distribution* which can achieve the best performance, $\varepsilon = 1$, in the ideal case.

Ideal soliton distribution $\rho(d)$:

$$\rho(d) = \begin{cases} \frac{1}{k} & \text{for } d = 1 \\ \frac{1}{d(d-1)} & \text{for } d = 2, 3, \dots, k \end{cases} \quad (1)$$

Ideal soliton distribution guarantees that all the release probabilities are identical to $1/k$ at each decoding step. Hence, there is exactly one expected ripple generated at each step when the encoding symbol size is k . After k processing step, ideally, the source data can be fully recovered.

However, ideal soliton distribution works poorly in practice. Belief propagation may be suspended by a small variance of the stochastic encoding/decoding situation in which no ripple exists because the expected ripple size is only one at any moment. According to the theory of random walk, the probability that a random walk of length k deviates from its mean by more than $\ln(k/\delta)\sqrt{k}$ is at most δ . It is a baseline of ripple size that must be maintained to complete the decoding process. Therefore, in the same paper by Luby, a modified version called *robust soliton distribution* was also proposed.

Robust soliton distribution $\mu(d)$:

$$S = c \ln(k/\delta) \sqrt{k},$$

$$\tau(d) = \begin{cases} S/dk & \text{for } d = 1, \dots, k/S - 1 \\ S \ln(S/\delta)/k & \text{for } d = k/S \\ 0 & \text{for } d = k/S + 1, \dots, k \end{cases}, \quad (2)$$

$$\beta = \sum_{d=1}^k (\rho(d) + \tau(d)), \quad (3)$$

$$\mu(d) = \frac{\rho(d) + \tau(d)}{\beta} \text{ for } d = 1, \dots, k, \quad (4)$$

where c and δ are two parameters for controlling the character of a robust soliton distribution. c controls the average degree of the distribution, and δ estimates that there are $\ln(k/\delta)\sqrt{k}$ expected ripples as aforementioned. Robust soliton distribution can ensure that only $n = k + \mathcal{O}(\ln^2(k/\delta)\sqrt{k})$ encoding symbols are required to recover the source data successfully with a probability at least $1-\delta$.

C. Evaluation of Degree Distributions

In this section, we outline two approaches to evaluate degree distributions. One intuitive indicator for evaluating degree distributions is the reception overhead, ε . For the reasons that the coding process of LT codes is stochastic and there is uncertainty in the transmission channel, a successful decoding cannot be guaranteed in the condition of a constant overhead. Moreover, in order to obtain the average overhead, a large amount of simulations are required to estimate ε . Therefore, an alternative approach is to evaluate the error probability of LT codes with some particular reception overhead. In 2004, an effective evaluation method was proposed in [10] for LT codes with a finite size of k . Dynamic programming was utilized to construct the distribution of ripple size during the decoding process. In short, the error probability of LT codes can be deterministically evaluated by the method when the input symbol size k and a fixed overhead are given. Unfortunately, the computation considers a 3-dimensional matrix of which the edge size is k and the complexity is too high. Even when several reduction techniques are applied, the computational complexity is still $\mathcal{O}(k^3 \log^2(k) \log \log(k))$. Subsequently, a new model for rigorous analysis on LT codes has been proposed in 2006 [11]. The difference in this approach is to make an assumption that the number of received symbols is a random variable with mean n . Based on the assumption, a fast evaluation with time complexity $\mathcal{O}(k^2 \log(k))$ was presented. Both approaches are feasible to serve as an evaluation function and are employed in this paper for different requirements.

III. PROBABILITY REALLOCATION

For seeking a reasonable strategy to select sparse tags, we conduct experiments to observe the effects of the probability reallocation on the degree distribution. We first define the probability reallocation process. For a given optimal degree distribution $\Omega(d)$, we choose degrees i, a , and b , where $a, b \neq i$ to apply reallocation and get the new degree distribution $\Omega'(d)$:

$$\Omega'(d) = \begin{cases} \Omega(d) & \text{for } 1 \leq d \leq k, d \neq i, a, b \\ 0 & \text{for } d = i \\ \Omega(d) + \Delta(d) & \text{for } d = a, b \end{cases},$$

where $\Delta(a) + \Delta(b) = \Omega(i)$, i.e. reallocate the probability of degree i to degrees a and b . After determining degrees i, a , and b , an evolutionary algorithm is applied to find out the best distribution of the probability reallocation on these degrees. The objective of the best distribution denotes the least increment of error probability after reallocation. In [5], a method that can find the optimal degree distributions for LT codes with small k size was presented. The obtained optimal degree distributions guarantee the minimal error probability for overhead $\varepsilon = 1$. Calculation of the error probability for LT codes at $\varepsilon = 1$ can be done by Karp's evaluation approach. Hence, the optimal degree distribution for input symbol size $k = 7$ was chosen as our experimental subject and the covariance matrix adaptation evolution strategy [12], [13], [14](CMA-ES) is employed to search for the best reallocation. In the experiment, we denote the minimal error probability

(a) Error probability variances

(b) Error probability variances in log scale

Fig. 1. Error probability variances after reallocation.

for input symbol size $k = 7$ as BP_7 and for reallocated distribution with pair (a, b) as $RP_{(a,b)}$. Figure 1(a) shows the differences of error probabilities between the optimal and the reallocated degree distributions.

In the experiment, we choose the optimal degree distribution for $k = 7$ as the observation target and redistribute the probability of degree 4 to any combinations of degree (a, b) with $a, b \neq 4$. The results illustrate two factors of error probability variances that: 1) the distance between the reallocated and the removed degrees; 2) the complementary property of the reallocated degrees. For the first factor, the figure shows that the differences of error probabilities would grow as the distances of a, b to 4 become larger. For instance, if we fix $a = 1$ and change b from 1 to 7 excluding 4, it can be found that the differences would be decreased when the value of b close to 4. For the latter, the complementary property means one of the pair (a, b) is bigger than the removed degree and the other is smaller would be better. It can be

Algorithm 1 Tags Selection Function

Input: The source symbol size k , the density parameter d ;**Output:** The set of sparse tags;

```
1: procedure TSF( $k, d$ )
2:    $D \leftarrow$  Ideal soliton distribution for size  $k$ ;
3:    $S \leftarrow 0, E \leftarrow 1, Tags \leftarrow []$ ;
4:   while  $i < k$  do
5:      $S \leftarrow S + E \times D(i)$ 
6:     if  $S > (1/d)$  then
7:        $Tags \leftarrow [Tags, i]$ ;
8:        $S \leftarrow 0$ ;
9:        $E \leftarrow 1$ ;
10:    else
11:       $E \leftarrow E + 1$ ;
12:    end if
13:  end while
14:  return vector  $[1, Tags, k]$ ;
15: end procedure
```

observed by comparing the selections of $(a, b) = (a_1, 4 + i)$ to $(a, b) = (a_1, 4 - i)$, e.g. comparisons between $(1, 2)$ and $(1, 6)$. The distance influence of such two pairs are the same but the pair $(1, 6)$ with complementary property observably has a smaller difference.

Given the above considerations, it turns out that the reallocation of the probability to degrees 3 and 5, the nearest two degrees to 4, would be the most close to the optimum. For a better view of showing how close the adjusted distributions could approach to the optimal one, we illustrate the differences of error probabilities in logarithm scale in Figure 1(b). It also gives the proof that the error probability of the original distribution could be well approximated by the reallocated one. These observations inspire us regarding how to choose the tags for a sparse degree distribution.

IV. SELECTION FUNCTION FOR SPARSE TAGS

In addition to the factors observed from experiments, we also take some intuitive properties into account. For example, the higher probability to be reallocated would result in higher difference of the error probability to the optimal one. Summing up all of above, the main considerations of our degree selection strategy for sparse tags would be as follows:

- 1) The number and value of probabilities around each degree.
- 2) The distance between the probability reallocated degrees and the removed one.

The first criterion comes from the positive correlation of the reallocated probability and the error probability. The second one accounts for the results of the experiments that replacing the tag to be removed with two adjacent tags would have the best approximation for error probability. According to these criterion, we propose the sparse tags selection function in Algorithm 1.

We consider the *ideal soliton distribution* since it would be the optimal degree distribution in the ideal case. The density

Fig. 2. Illustration of the work done by TSF(k, d).

parameter d acts as the bound that base on the first main consideration, degree i would not be removed if its probability was larger than $1/d$. On the other hand, we group the degrees with probabilities below $1/d$ and concentrate those probabilities to a nearby degree. The second main consideration would be applied to the selection of the representative degree of each group. We accumulate the probabilities multiplied the distance factors and take the degree while the sum exceeds the bound $1/d$. In addition, considering the complementary property of the selected degrees to distribute the probability, we reserve the degrees 1 and k to ensure there always exists degrees satisfied this property to be chosen. Figure 2 illustrates the work done by Tags Selection Function (TSF) and shows the tags selected for $(k, d) = (30, 10)$. Tags 1 and 30 were selected to meet the complementary property. The tags 2, 3 were selected since the probabilities of these tags in ideal soliton distribution were above the density criteria. The remaining tags were the representations of the grouped tags of which probabilities were below the density bound.

Following we provide some examples of sparse tags selected by Algorithm 1 for $k = 100$:

- 1) TSF(100, 1) = [1, 4, 23, 100].
- 2) TSF(100, 3) = [1, 2, 5, 12, 30, 76, 100].
- 3) TSF(100, 5) = [1, 2, 4, 8, 16, 32, 64, 100].
- 4) TSF(100, 10) = [1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 100].
- 5) TSF(100, 20) = [1, 2, 3, 4, 6, 8, 11, 15, 21, 29, 40, 56, 78, 100].

We can see that the sparse tags selected in TSF(100, 5) were close to the series of power of 2 and those selected in TSF(100, 10) were close to Fibonacci series. Such a result gives an explanation for the good performance of choosing these series, power of 2 and the Fibonacci series, as approximation to the full tags in a certain extent.

TABLE I
THE ERROR PROBABILITIES OF THE BEST DEGREE DISTRIBUTIONS DISCOVERED IN EACH 30-RUN EXPERIMENT.

Tag Type	k = 100	k = 150	k = 200	k = 250	k = 300
d = 3	0.56332720	0.46689819	0.37998428	0.30450876	0.24139928
d = 5	0.56296776	0.46627322	0.37872813	0.30325441	0.24007702
d = 10	0.56293832	0.46622717	0.37871451	0.30324585	0.24008108
d = 20	0.56291529	0.46619257	0.37866506	0.30319090	0.24001890
Full Tags	0.56291403	0.46619313	0.37869051	0.30323273	0.24023868
Min.	0.56291403	0.46619257	0.37866506	0.30319090	0.24001890

(a) Results in early stages for observing different initializations

(b) Results for sparse degree distributions converge faster

Fig. 3. The evolutionary trends of fitness values in optimization.

V. EXPERIMENTAL RESULTS

In this section, we make a complete examination on the effects of the proposed sparse tags selection function. We set up the input parameters $k = \{100, 150, 200, 250, 300\}$ and $d = \{3, 5, 10, 20\}$. For each (k, d) pair, we firstly use the TSF function to determine the corresponding tags and then apply the CMA-ES algorithm to search for the optimal sparse degree distribution with these tags. Finally, we compare the minimal error probabilities of the sparse degree distribution and the full degree distribution. For each optimization setting, 30 independent runs were tested due to the natural

randomness of evolutionary algorithms. The minimal error probability in 30 runs was recorded for each generation. The maximum number of function evaluation is limited to 3×10^4 and the optimization result is presented in Figure 3. Figure 3(a) shows the early phase of optimization process. The results of all sparse degree distributions are with high error rates at beginning because the initial probabilities on sparse degrees were set by random values. In contrast, the full degree distribution was initialized as ideal soliton distribution to avoid the failure of optimizing a large number of decision variables. As the number of evaluations increases, the curves of sparse distributions drop down to the same level of error probability as the full degree distribution. Since the density parameter, d , of our selection function effects the size of tags, it also reflects on the convergence of the curve of each sparse distribution. It can be observed that the curve with $d = 3$ firstly converges and $d = 20$ is the last. Figure 3(b) shows the same experimental result but in a different interval of x -axis. We can clearly observe that under the same optimization approach, the optimized results of sparse distributions are better than that of the full degree after hundreds of function evaluations.

Fast convergence is an expected advantage for adopting sparse degree distributions. Furthermore, it can be expected that the tags defined by our selection strategy could approximate full degrees on performance as nicely as possible. To examine our argument, the minimal error probabilities of different tag types are listed in Table I for comparison. The values were the minimal results found by CMA-ES in 30 runs and 3×10^4 function evaluations. For each column, the minimal error probability is marked as bold and copied to the last row. The sparse degree distribution with $d = 20$ leads four of totally five different k size experiments. For giving a more convenient view to compare the results, we survey the differences between each entry and the minimal value in same column. Accounting for the minimal error probability of the column will be zero after all entries minus the minimal one, we add a base value (10^{-7}) to calculate differences for letting all data be plotted in a logarithmic coordinate system. Figure 4 visualizes the values in Table I and shows the distance between each tag types and the near optimal distribution that we have found. Although a larger value of d which means a larger subset of degrees will cause slow convergence in optimization, more tags can form a sparse distribution with a lower error probability. Our experimental result definitely confirms the argument and also illustrates that the selection strategy is practical. On the other hand, the full degree distribution is

Fig. 4. Differences between the error probabilities and the minimal one.

considered to have global minimal error probability because it is the universal set of all distributions and forms the complete search space. However, the optimization result of full degree distributions gets worse and worse as input symbol size k increases. For the same optimization algorithm and evaluation function are implemented for each degree set, the worse results of full degree distributions could be explained by that the number of decision variables is too many for CMA-ES to handled in limited function evaluations.

VI. CONCLUSION

Using evolutionary algorithms to optimize the degree distribution for LT codes is a promising research topic. Sparse degree distributions are frequently used to replace full degrees for reducing the search space. How good the performance could be achieved by a sparse degree distribution depends on the set of its non-zero entries, i.e., tags. However, no investigation has been done regarding how to decide appropriate tags to construct sparse degree distributions with good performance. In this paper, the authors analyzed the influence of different degrees on decoding rate and proposed a tag selection algorithm to choose tags for LT codes optimization. Finally, the presented experimental results were evidentially illustrate the practicality of the proposed tag selection algorithm.

In previous studies, researchers manually chose tags for sparse degree distributions according to their own experimental experience. Even though the chosen subset of degrees worked well, the detailed mechanism was still unknown. This work made an effort to find out guidelines for choosing appropriate tags. The proposed selection algorithm can be applied for any input size and control the level of sparseness conveniently by adjusting the density parameter. This solution can help researchers to put more attention in the optimization algorithm rather than the individual encoding. The paper presented the qualitative analysis of probability reallocation in a distribution.

The variances of error probability were compared for changing the reallocated degree and then quantitative analysis is needed in advance. If the quantity of variance can be measured precisely, developing a local search based on the measure approach to enhance certain optimization framework for LT codes will be possible. Research of this line is definitely worth pursuing, and the authors are currently taking the challenge.

ACKNOWLEDGMENTS

The work was supported in part by the National Science Council of Taiwan under Grant NSC 99-2221-E-009-123-MY2.

REFERENCES

- [1] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 1998, pp. 56–67.
- [2] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1528–1540, 2002.
- [3] M. Luby, "LT codes," in *Proceedings of the 43rd Symposium on Foundations of Computer Science*. IEEE Computer Society, 2002, pp. 271–282.
- [4] E. A. Bodine and M. K. Cheng, "Characterization of Luby Transform codes with small message size for low-latency decoding," in *Proceedings of the IEEE International Conference on Communications*, 2008, pp. 1195–1199.
- [5] E. Hytiä, T. Tirronen, and J. Virtamo, "Optimal degree distribution for LT codes with small message length," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, 2007, pp. 2576–2580.
- [6] —, "Optimizing the degree distribution of LT codes with an importance sampling approach," in *Proceedings of the 6th International Workshop on Rare Event Simulation (RESIM 2006)*, 2006, pp. 64–73.
- [7] C.-M. Chen, Y.-p. Chen, T.-C. Shen, and J. K. Zao, "On the optimization of degree distributions in LT code with covariance matrix adaptation evolution strategy," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2010, pp. 3531–3538.
- [8] —, "Optimizing degree distributions in LT codes by using the multiobjective evolutionary algorithm based on decomposition," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2010, pp. 3635–3642.
- [9] A. Talari and N. Rahnavard, "Rateless codes with optimum intermediate performance," in *Proceedings of the Global Telecommunications Conference (GLOBECOM 2009)*, 2009, pp. 1–6.
- [10] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT codes," in *Proceedings of the IEEE International Symposium on Information Theory 2004 (ISIT 2004)*, 2004, p. 39.
- [11] E. Maneva and A. Shokrollahi, "New model for rigorous analysis of LT-codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2006)*, 2006, pp. 2677–2679.
- [12] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.
- [13] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, 2005, pp. 1777–1784.
- [14] —, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, 2005, pp. 1769–1776.