Optimization of Unequal Error Protection AL-FEC Scheme Via Adapted Simulated Annealing

César Díaz, Julián Cabrera, Fernando Jaureguizar, and Narciso García

***Abstract*—We adapt simulated annealing for speeding up the search of optimal configurations for protecting video transmission over IP networks. The considered protection scheme is a version of the Pro-MPEG COP3 FEC codes, consisting in using several matrices of unequal size per FEC block.**

I. Introduction

In scenarios where video transmission services are supplied through managed IP networks (e.g. television broadcasting, video on demand or videoconferencing), Application-Layer Forward Error Correction (AL-FEC) techniques are commonly introduced for increasing the reliability of the communication channel [1], [2]. In most of the cases, these services demand very low latency for ensuring good performance and satisfying the user's quality expectations. For this reason, protection mechanisms must meet the imposed real-time conditions [3].

In this paper, we consider an unequal error protection (UEP) version of the broadly-used Pro-MPEG COP3 AL-FEC codes introduced by the Pro-MPEG Forum in its Code of Practice 3 r2 [4]. This version, proposed in [5], enables the use of a number of matrices of dissimilar size per protection block, so that unequal code rates can be applied to different groups of data packets in regard of their importance in terms of the potential degradation that their loss might cause. With the purpose of using the most suitable matrix configuration (i.e., the number of matrices and their size with which the decoded sequence is estimated to end up less distorted), an optimization process needs to be launched within every block. This process, however, might take too long if brute-force search algorithms are used, especially if the number of data and repair packets per block is high, since the number of possible configurations increases alongside.

Given the characteristics of the problem, we resort to simulated annealing (SA) to speed up the optimization process. SA is a general probabilistic tool for solving nonlinear optimization problems that has been successfully applied in a wide range of areas [6], [7], traditionally to make problems with a large amount of solutions manageable. Furthermore, as SA is highly prone to problem-specific adaptations, we use it to make the above mentioned procedure fast and robust enough to be included in commercial transmission appliances.

This work has been partially supported by the Ministerio de Economía y Competitividad of the Spanish Government under project TEC2013-48453 (MR-UHDTV).

II. Problem Description and Proposed Approach

The problem is formalized as follows. Let N_M be the number of matrices in a configuration, and C_m and R_m respectively the number of columns and rows in matrix $m, 1 \leq m \leq N_\mathrm{M}$. We assume that matrix 1 protects the $C_1 \cdot R_1$ most relevant packets in the protection block, matrix 2 the following $C_2 \cdot R_2$, and so on. The relevance of the different packets is the result of applying a specific distortion model to the packet stream. This model is out of the scope of this paper. The goal of the optimization problem is to find the most convenient values of the variables N_M, $C_1, \ldots, C_{N_\mathrm{M}}$, $R_1, \ldots, R_{N_\mathrm{M}}$ to protect the N_P data packets in the block. The problem is subject to the following conditions:

1) All data packets are protected, the first $N_\mathrm{M}-1$ matrices are full and no row is left empty
$$\sum_{m=1}^{N_\mathrm{M}-1} C_m \cdot R_m + C_{N_\mathrm{M}} \cdot (R_{N_\mathrm{M}} - 1) < N_\mathrm{P} \leq \sum_{m=1}^{N_\mathrm{M}} C_m \cdot R_m$$
2) Assuming an essentially bursty channel, parity packets are generated only column-wise
$$\sum_{m=1}^{N_\mathrm{M}} C_m = N_\mathrm{FEC}$$
where $N_\mathrm{FEC} = N_\mathrm{P} \cdot ((1/r_\mathrm{FEC}) - 1)$ is the number of repair packets, where r_FEC is the imposed minimum code rate.

Assumed the first condition, the size of one of the matrices can be obtained from that of the others. The dimensions of all the matrices but one then are the $2 \cdot (N_\mathrm{M} - 1)$ degrees of freedom of the system. The number of possible combinations of values that fulfill both restrictions can be quite vast, particularly for high N_P, N_M, and low r_FEC.

In our approach, the first step considered for accelerating the optimization process is delimiting the solution space. To that end, the next restrictions are included:

3) More resources devoted to more important packets
$$C_{m_1} \geq C_{m_2}, 1 \leq m_1 \leq m_2 \leq N_\mathrm{M}$$
4) Lower code rates devoted to more important packets
$$C_{m_1} \cdot R_{m_1} \leq C_{m_2} \cdot R_{m_2}, 1 \leq m_1 \leq m_2 \leq N_\mathrm{M}$$

Those conditions make the number of configurations go down, barely affecting the result of the process. Thus, the SA procedure starts from a more advantageous situation.

TABLE I
RESULTS OF THE EXPERIMENTS IN TERMS OF NUMBER OF CONSIDERED CONFIGURATIONS, PROCESSING TIME AND ERROR

		num. configurations			time (s)			error (%)		
Approach	N_M	Seq1	Seq2	Seq3	Seq1	Seq2	Seq3	Seq1	Seq2	Seq3
Brute-force search (all solutions)	2	824	2536	3440	0.0115	0.1449	0.2485	-	-	-
	3	292460	2929004	5433062	3.5362	165.403	469.172	-	-	-
	4	61372536	594327460	1023585461	926.788	41069.3302	92809.481	-	-	-
Brute-force search (restricted solutions)	2	71	186	238	0.0015	0.0073	0.01	0	0	0
	3	2209	13267	21842	0.03517	0.4378	0.9486	0	0	0
	4	33787	488963	1022504	0.5448	29.1326	82.2587	1.79	0	0
Adapted SA algorithm (restricted solutions)	2	71	186	238	0.0017	0.0037	0.0054	0	0	0
	3	2209	13267	21842	0.0126	0.0385	0.0749	2.64	0	0
	4	33787	488963	1022504	0.0451	1.099	2.5477	2.38	0.01	0.37

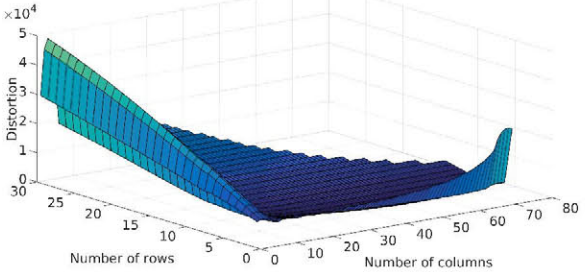

Fig. 1. Subset of the solution space for $N_M = 2$ (possible combinations of dimensions of matrix 1), and distortion associated

The second step of our approach is based on the fact that, in our problem, near solutions are highly correlated and, despite not constituting a convex set of points, the distance between the solutions and the convex hull containing this set is very short. That can be checked in the example depicted in Fig. 1. For taking advantage of this characteristic, we adapt the SA algorithm, so that jumping to a new solution not only depends on the temperature and the cost of the solutions, but also on the distance between them. The distance between configurations $s_1 = C_1^{s_1}, R_1^{s_1} ... C_{N_M}^{s_1}, R_{N_M}^{s_1}$ and $s_2 = C_1^{s_2}, R_1^{s_2} ... C_{N_M}^{s_2}, R_{N_M}^{s_2}$ of N_M matrices is defined as:

$$d_{N_M}(s_1, s_2) = \left\{ \sum_{m=1}^{N_M - 1} \left[(C_m^{s_1} - C_m^{s_2})^2 + (R_m^{s_1} - R_m^{s_2})^2 \right] \right\}^{\frac{1}{2}} \quad (1)$$

At each iteration i, we only consider the solutions that are separated less than a decreasing maximum distance $d_{N_M}(i)$ from the current one. This distance is computed as:

$$d_{N_M}(i) = d_{N_M}^{\text{init}} \cdot \beta^i \quad (2)$$

where $d_{N_M}^{\text{init}}$ is the distance initial value, and $0 < \beta < 1$. In this way, (i) we favor downhill transitions to some degree, without compromising the capacity of the algorithm to avoid local minima; and (ii) the number of solutions to be evaluated highly decreases with i, resulting in an acceleration of the process at a potential cost of a very small, acceptable error.

III. EXPERIMENTS AND CONCLUSION

With the aim of evaluating our approach, we have employed three sequences of 2.5 Mbps (Seq1), 4 Mbps (Seq2) and 8 Mbps (Seq3). FEC blocks have been selected to accommodate the data packets corresponding to one second of video each time, which averagely means sets of approximately 230, 369, and 737 packets, respectively. Moreover, r_{FEC} equals 0.91. Thus, the number of repair packets per block approximately is 23, 37, and 74, respectively. Finally, we have chosen $d_{N_M}^{\text{init}} = (N_M - 1) \cdot N_{\text{FEC}}$ and $\beta = 0.5$.

The two proposed steps are assessed in terms of average number of configurations considered, processing time (with a 2-core GPU clocked at 3 GHz with 12 GiB RAM), and error. The latter measures the difference between the optimal solution and the one reached regarding the introduced distortion.

The experiment results are shown in Table I. Regarding the first step, we can clearly see the great decrease in the number of solutions to be considered when applying the restrictions mentioned before, which clearly implies an enormous reduction in terms of computing time. The error introduced corresponds to the rare cases where the optimal solution is ruled out of the solution space finally used. Nevertheless, its value is very low, as quasi-optimal solutions are reached.

If we now consider the second step, we see a considerable speed-up when opting for the proposed adaptation of the SA algorithm, which in most of the cases makes the protection scheme completely suitable for real-time applications, at the expense of a very low error. Moreover, this has been achieved even without considering extra acceleration techniques (e.g. parallelization) or paying any particular attention to the initialization process of the algorithm, which is so far carried out randomly.

REFERENCES

[1] M. Luby, T. Stockhammer, and M. Watson, "IPTV systems, standards and architectures: Part II - Application Layer FEC in IPTV services," *IEEE Communications Magazine*, pp. 94–101, May 2008.

[2] F. Battisti, M. Carli, E. Mammi, and A. Neri, "A study on the impact of AL-FEC techniques on TV over IP Quality of Experience," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 86, 2011.

[3] ETSI TS 102 542-3-2, "Digital Video Broadcasting (DVB): Guidelines for the implementation of DVB-IP Phase 1 specifications. Part 3: Error Recovery. Subpart 2: Application Layer FEC," May 2011.

[4] Pro-MPEG Forum Code of Practice 3 release 2, "Transmission of professional MPEG-2 transport streams over IP networks," Jul. 2004.

[5] C. Díaz, J. Cabrera, F. Jaureguizar, and N. García, "An Extention to the Pro-MPEG COP3 Codes for Unequal Error Protection of Real-time Video Transmission," in *22th IEEE Int. Conf. on Image Proc. (ICIP)*, Sep. 2015.

[6] Z. Shi, W. Fernando, and A. Kondoz, "Adaptive direction search algorithms based on motion correlation for block motion estimation," *IEEE T. on Consumer Electronics*, vol. 57, no. 3, pp. 1354–1361, Aug. 2011.

[7] M. Hosen, A. Khosravi, S. Nahavandi, and D. Creighton, "Improving the Quality of Prediction Intervals Through Optimal Aggregation," *IEEE T. on Industrial Electronics*, vol. 62, no. 7, pp. 4420–4429, Jul. 2015.