References

1 PLESS, V.: 'Decoding the Golay codes', *IEEE Trans.*, 1986, **IT-32**, (4), pp. 561–567
2 VARDY, A., and BE'ERY, Y.: 'More efficient soft decoding of the Golay codes', *IEEE Trans.*, 1991, **IT-37**, (3), pp. 667–672
3 MACWILLIAMS, F.J.: 'Self-dual codes over GF(4)', *J. Comb. Theory A*, 1978, **25**, (3), pp. 288–318
4 WOLF, J.: 'Efficient maximum likelihood decoding of linear block codes using a trellis', *IEEE Trans.*, 1978, **IT-24**, (1), pp. 76–80

Weakly constrained codes

K.A.S. Immink

Indexing term: Constrained codes

The author reports on the performance of a new class of constrained codes, called weakly constrained codes. These codes do not strictly guarantee the imposed channel constraints, but rather generate codewords that violate, with a given (small) probability, the prescribed constraint. Weakly constrained codes are specifically of interest when it is desirable that the code rate $R = p/q$ is very high, requiring codewords of length $q > 100$.

Introduction: Codes for the noiseless constrained channel, often called *constrained codes* convert the source sequence into a signal suitable for specific physical requirements. Examples are the run-length-limited (RLL) and DC-balanced codes, which have found widespread application in storage products and cable transmission systems [1]. RLL or (d, k) codes generate sequences having at least d, and at most k, 'zeros' between consecutive 'ones'.

In coding practice, the source sequence is partitioned into blocks of length p, and under the code rules, such blocks are mapped onto words of q channel symbols. The rate of such an encoder is $R = p/q \leq C$, where C is the noiseless capacity of the constrained channel [2]. For certain applications it is desirable that the code rate $R = p/q$ is very high, requiring a codeword length $q > 100$. The construction of such high-rate codes is far from obvious, as table look-up for encoding and decoding is an engineering impracticality.

The publications by Fair *et al.* [3] and Immink and Patrovics [4] on *guided scrambling* brought new insights into high-rate code design. Guided scrambling is a member of a larger class of related coding schemes called *multimode* codes. In multimode codes, the p bit source word is mapped into $(m+p)$ bit codewords. Each source word can be represented by a member of a *selection set* consisting of $L = 2^m$ codewords. Examples of such mappings are the guided scrambling algorithms presented by Fair *et al.* [3], the DC-free coset codes of Deng and Herro [5], and the scrambling using a Reed-Solomon code by Kunisa *et al.* [6]. A mapping is considered to be 'good' if the selection set contains sufficiently distinct and random codewords. Provided that 2^m is large enough and the selection set contains sufficiently different codewords, multimode codes can also be used to satisfy almost any channel constraint with a suitably chosen selection method.

In the context of high-rate multimode codes, there is a growing interest in *weakly constrained* codes. Weakly constrained codes fail to comply with the rules as they produce sequences that violate the constraints with (low) probability p. It is argued that if the channel is not free of errors, it is pointless to feed the channel with perfectly constrained sequences. The following Section describes weakly constrained $(0, k)$ codes.

Example: Each source word of length p, $p \gg 1$, is supplemented by m bits, so that the codeword length is $q = p+m$. The rate of the code is $R = p/q$. The m supplement bits make it possible to generate a selection set of $L = 2^m$ q-sequences. The particular method for generating the selection set is not discussed; we merely assume that the the selection set comprises sufficiently distinct and random words. We assume that we have a channel constraint which limits the channel capacity to C; then the number of constrained codewords can be approximated by [2]

$$N(q) \simeq A2^{qC} \qquad (1)$$

where A is a constant independent of q. The probability that in L drawings from randomly generated sequences we will not find a single sequence obeying the given constraint is simply

$$p = (1 - p_0)^L \qquad (2)$$

where

$$p_o = \frac{N(q)}{2^q} \simeq A2^{(C-1)q} \qquad (3)$$

As $L = 2^m$ and $m = (1-R)q$, we have

$$p = \left(1 - A2^{(C-1)q}\right)^{2^{(1-R)q}} \qquad (4)$$

If, for simplicity, it is assumed that $A2^{(C-1)q} \ll 1$, we have

$$\ln(p) = -A2^{(C-R)q} \qquad (5)$$

(0,k) codes: The number of $(0, k)$ sequences that start and end with at most $l = \lfloor k/2 \rfloor$ and $r = k-l$ 'zeros', is given by

$$N_c(q) \simeq A_c 2^{qC(0,k)} \qquad (6)$$

where $A_c \simeq 1$ [7], and $C(d, k)$ is the capacity of the (d, k)-constrained channel. The capacity $C(0, k)$ equals $\log_2(\lambda)$, where λ is the largest root of [1]

$$x^{k+2} - 2x^{k+1} + 1 = 0 \qquad (7)$$

For a sufficiently large k, we derive

$$\lambda \simeq 2\left(1 - \frac{1}{2^{k+2}}\right)$$

so that

$$C(0,k) \simeq 1 - \frac{1}{\ln 2} 2^{-k-2} \qquad k \gg 1 \qquad (8)$$

Combining the above with eqn. 5, we find that

$$\ln(p) = -2^{(C-R)q} \qquad (9)$$

Fig. 1 shows the probability p that no sequence taken from a selection set of random sequences of size $L = 2^m$ obeys the $(0, k)$ constraint.

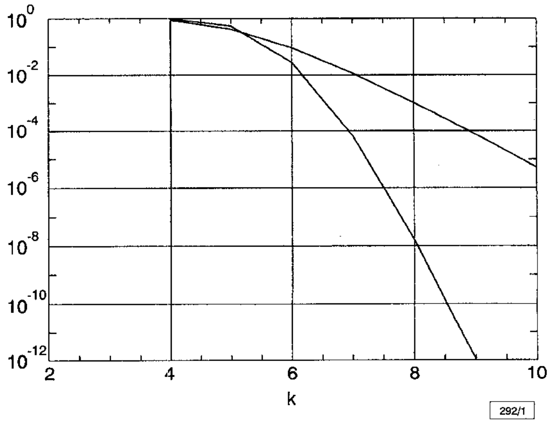

Fig. 1 *Probability that no sequence of L drawings from a selection set of random sequences satisfies (0, k) constraint*
Code rate $R = 0.99$
Upper curve: codeword length $q = 200$, selection set size $L = 4$
Lower curve: codeword length $q = 400$, selection set size $L = 16$

Let the code rate $R = 99/100$ and the codeword length $q = 400$, then the size of the selection set is $L = 16$. We observe that with probability $P = 10^{-12}$, a codeword violates the $k = 9$ constraint. The alternative impleluentation [8] requires a rate of $R = 24/25$, four times the redundancy of the weakly constrained code, to strictly guarantee the same (0,9) constraint.

Conclusions: We have studied the performance of weakly constrained codes. Weakly constrained codes do not strictly guarantee the imposed channel constraint but rather generate codewords that violate with (small) probability p the prescribed constraint. It has been shown that $(0, k)$ weakly constrained codes of rate 0.99, generate with probability $P = 10^{-12}$ a codeword that violates the $k = 9$ constraint.

© IEE 1997 1 September 1997
Electronics Letters Online No: 19971361

K.A.S. Immink (*Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands*)

E-mail: immink@natlab.research.philips.com

References

1 IMMINK, K.A.S.: 'Coding techniques for digital recorders' (Prentice-Hall International (UK) Ltd., Englewood Cliffs, New Jersey, 1991)

2 SHANNON, C.E.: 'A mathematical theory of communication', *Bell Syst. Tech. J.*, 1948, **27**, pp. 379–423

3 FAIR, I.J., GOVER, W.D., KRZYMIEN, W.A., and MACDONALD, R.I.: 'Guided scrambling: A new line coding technique for high bit rate fiber optic transmission systems', *IEEE Trans. Commun.*, 1991, **COM-39**, (2), pp. 289–297

4 IMMINK, K.A.S., and PATROVICS, L.: 'Performance assessment of DC-free multimode codes', *IEEE Trans. Commun.*, 1997, **45**, (3), pp. 293–299

5 DENG, R.H., and HERRO, M.A.: 'DC-free coset codes', *IEEE Trans. Inf. Theory*, 1988, **IT-34**, (4), pp. 786–792

6 KUNISA, A., TAKAHASHI, S., and ITOH, N.: 'Digital modulation method for recordable digital video disc', *IEEE Trans. Consum. Electron.*, 1996, **42**, pp. 820–825

7 IMMINK, K.A.S.: 'A practical method for approaching the channel capacity of constrained channels', *IEEE Trans. Inf. Theory*, 1997, **43**, (5), pp. 1389–1399

8 IMMINK, K.A.S., and VAN WIJNGAARDEN, A.: 'Simple high-rate constrained codes', *Electron. Lett.*, 1996, **32**, (20), pp. 1877

Joining of compacted cells using genetic algorithm

A. Lim and Hoe-Kit Chew

Indexing terms: Genetic algorithms, VLSI

The authors study the minimum joining area for k compacted cells in a row using a genetic algorithm. The genetic algorithm approach has obtained optimal results in many test cases and yielded significantly better results compared to greedy and local search algorithms.

Introduction: In [2], various algorithms for the joining of two compacted cells were considered. Lim *et al.* [3] proposed a constraint graph model for representing the cell joining of k compacted cells in a row. In their model, if the number of tracks to be used in each channel is known, the minimal joining area for dependent stretching can be found in $O(n)$ time. Unfortunately, the general problem of area minimisation is NP-complete. In this Letter, we adapted the graph model proposed in [3] and applied genetic algorithm search techniques to find the minimium joining area. We compare the results obtained using a genetic algorithm with those obtained with other techniques.

Consider the joining problem for two cells, L and R. Cell L is to the left of cell R and each cell has n terminals. The terminals of L are on its right vertical boundary while those of R are on its left vertical boundary. Let T_i^A denote terminal i of cell A, $A \in \{L, R\}$. Let p_i^A, $A \in \{L, R\}$ be the position of T_i^A, $1 \le i \le n$. Let $p_0^A = 0$ and $p_{n+1}^A = h_A$, where h_A is the height of cell A. Since we assume a virtual grid system, all the $n+2$ values of p_i^A are integers. Let $\Delta_i^A = p_{i+1}^A - p_i^A$, $0 \le i \le n$. Stretching cell A is equivalent to repositioning the terminals of A on the virtual grid line corresponding to the right edge of A in the case $A = L$ and left edge of A in case $A = R$. Let q_i^A be the position of terminal i of A following stretching, $1 \le i \le n$. Let $q_0^A = 0$ and $q_{n+1}^A = q_n^A + \Delta_n^A$. Then, the height of the stretched cell A is q_{n+1}^A. The values of q_i^A define a *legal stretching* of cell A iff $q_{i+1}^A - q_i^A \ge \Delta_i^A$, $0 \le i \le n$.

For any set $Q = \{q_i^A | 1 \le i \le n, A \in \{L, R\}\}$ of terminal positions, the terminal pairs (q_i^L, q_i^R), $1 \le i \le n$ can be river routed using s, $s \ge 0$ tracks iff $q_i^L \ge q_{i-s}^R + s$, $s < i \le n$ and $q_i^R \ge q_{i-s}^L + s$, $s < i \le n$ [1].

The joining problem for k cells is similar to the two cell case, except that there are k cells, $C = \{C_1, ..., C_k\}$ placed in a row to be joined together. With k cells, cell C_i is on the left of cell C_{i+1}, $1 \le i \le k - 1$. There are $k - 1$ channels where cell joinings are performed. The joining area is the sum of all the cell widths and channel widths multiplied by the height of the tallest cell. The area formula is given as $A = \max_{i=1}^{k} h_i \times \{\Sigma_{i=1}^{k} w_i + \Sigma_{i=1}^{k-1} c \times \min\{s_i, 1\} \times (s_i + 1)\}$, where h_i and w_i are the height and width of cell C_i, respectively; c is a technology dependent factor; s_i is the number of tracks needed to join C_i and C_{i+1}. If no tracks are needed, the channel width is 0, but, if j tracks are needed, the channel width is $(j + 1) \times c$, $j > 0$.

Given the channel densities, we can stretch the height of the cell such that the terminals of the cells can be routed. We will consider the situation where the stretching is such that if a terminal on one side is shifted up by x, all terminals of the cell located higher than that terminal are shifted up by x.

Graph model: Cut lines are defined as horizontal lines across the cell through the terminals as well as across the top and bottom of the cell. Let c_j denote the cut lines and the distance between c_j and c_{j+1} be d_j. For each cell, every cut line c_j is transformed to two vertices c_j^b and c_j^t. There are directed edges $<c_j^b, c_j^t>$ and $<c_j^b, c_{j+1}^t>$ with weights 0 and h_j, respectively. For each cell, a chain like that described above is generated. For k cells, $C_1, C_2, ..., C_k$, k chains will be produced. Between cells C_i and C_{i+1} is the routing channel i. Assume that s_i tracks are available for routing in channel i. Let L_j^i be the set of terminals on the right side of the cell C_i, and R_j^{i+1} be the set of terminals on the left side of the cell C_{i+1}. L_j^i are the terminals on the left of channel i and R_j^{i+1} are the terminals on the right of channel i. Let n_i be the number of these terminals. Each L_j^i is to be connected to R_j^{i+1}, $1 \le j \le n_i$.

These terminals can be river-routed using s_i tracks iff $L_j^i \ge R_{j-s_i}^{i+1} + s_i$, $s_i < j \le n_i$ and $R_j^{i+1} \ge L_{j-s_i}^i + s_i$, $s_i < j \le n_i$. This implies that L_j^i must be at least s_i units higher than $R_{j-s_i}^{i+1}$ and R_j^{i+1} must be s_i units higher than $L_{j-s_i}^i$. These constraints can be captured by an edge from the vertex representing the cut line through $R_{j-s_i}^{i+1}$ to the vertex representing the cut line through L_j^i and adding an edge from the vertex representing the cut line through $L_{j-s_i}^i$ to the vertex representing the cut line that cuts across through R_j^{i+1} with edge weight s_i. The addition of these edges across the chains is carried out for all terminals $< R_{j-s_i}^{i+1}, L_j^i >$ and $< L_{j-s_i}^i, R_j^{i+1} >$ for all $s_i < j \le n_i$, and for all channels i, $1 \le i < k$. In addition, we add a source vertex, S and a sink vertex, T. A directed edge with weight 0 from S is added to the vertex c_1^t of each chain. A directed edge from the top-most vertex from each chain to the sink vertex T, with weight zero is also added. The resultant graph is a directed acyclic graph (DAG) and the longest path in a DAG can be found in linear time. The longest path in the DAG from S to T, which is equivalent to the minimum joining height, will give us the minimum joining area.

Benchmarking algorithms: Greedy algorithm (i): First, we initialise all channel densities to zero. Starting from the first channel, we hold all other channel densities constant while the channel density of the first is exhaustively tested to determine the best channel density for the first channel. This is then done to all subsequent channels. The area determined using this method is then the minimum joining area. This algorithm is used to generate the first set of benchmarks.

Greedy algorithm (ii): Let D_c denote the best-so-far channel densities. D_c is initialised to zero. We then vary each channel by one. If any variation results in a better area joining, D_c is updated. The algorithm terminates when no variations produce a better area joining.

Semi exhaustive algorithm: All our test data is for various cell sizes has a maximum channel density of 16 in the worst case. For such problems, it has been found that very good solutions have channel densities such that $0 \le s_i \le 3$. For this reason, we have decided to determine the minimum joining area for a subset of all combinations, i.e. for all s_i, $0 \le s_i \le 3$. This algorithm generates the third set of benchmarks.

Genetic algorithm: Each channel density s_i, maps to a gene. The string of genes make up the chromosome. The fitness of the chromosome is the joining area resulting from it's gene values. We