On the Construction of Turbo Code Interleavers Based on Graphs with Large Girth

Pascal O. Vontobel
ISI, ETH Zurich
Sternwartstr. 7
CH-8092 Zurich, Switzerland

***Abstract*—We discuss how interleavers for parallel concatenated turbo codes with good minimum distance can be derived from graphs having large girth, i.e. graphs whose length of the shortest cycle is large.**

***Index Terms*—Turbo codes, interleaver design, graphs with large girth.**

I. Introduction and Motivation

In [1] it is shown for parallel concatenated turbo codes that the minimum distance has an upper bound which is proportional to the logarithm of the interleaver length and therefore also proportional to the logarithm of the block length (see [1] for the exact expressions). In this statement it is tacitely assumed that we consider a class of turbo codes where the component convolutional codes are fixed and only the interleaver changes with the interleaver length. The idea of the proof is essentially that certain non-zero low-weight codewords can easily be shown to exist and one can give an upper bound on their weight. Our approach in this paper is to construct interleavers which try to avoid all these low-weight codewords as far as possible. We will derive the interleavers from graphs which have large girth (the girth of an undirected graph is the length of the smallest cycle).

Graphs (especially Cayley graphs) with large girth have been used for the construction of regular low-density parity-check (LDPC) codes in [2] [3] [4], irregular LDPC codes in [5] and regular LDPC codes with more complicated subcodes in [6]. For other algebraic constructions of interleavers based on other principles, see e.g. [7] or [8] and the references therein. See also [9] for some extensions of the results of [1].

The paper is structured as follows. In Sec. II we introduce different graphs representing parallel concatenated turbo codes and Sec. III discusses which low-weight codewords should be avoided. In Sec. IV we describe a class of graphs that will be used as building blocks for the interleaver construction. Sec. V presents different approaches for constructing interleavers and finally in Sec. VI we make some conclusions and state some open problems.

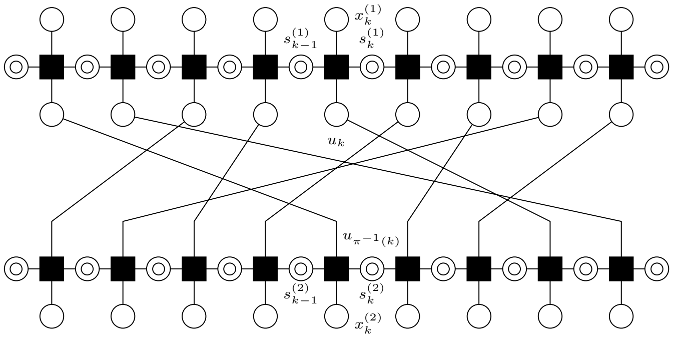

Fig. 1. Factor graph of a parallel concatenated turbo code with interleaver length 9.

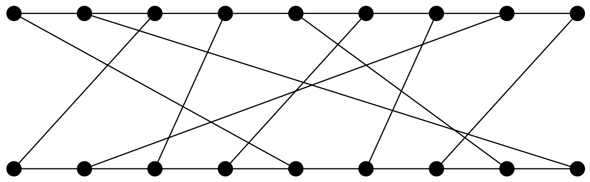

Fig. 2. Interleaver graph (IG) of a parallel concatenated turbo code with interleaver length 9.

II. Different Graphs representing Turbo Codes

In this section we discuss different representations of parallel concatenated turbo codes with the help of graphs. Parallel concatenated turbo codes consist of two systematic recursive convolutional codes (RCCs) whereby the input bits of the second RCC are the permuted input bits of the first RCC. A first possible representation is by factor graphs [10] [11] as given in Fig. 1. The empty simple circles represent input and output bits, the empty double circles represent states and the filled squares represent the constraints (trellis sections). The upper part represents the first RCC, the lower part the second RCC and inbetween is the interleaver which permutes the input bits. A typical constraint node (trellis section) of the first RCC has at time index k the state $s_{k-1}^{(1)}$ on the left hand side, the state $s_k^{(1)}$ on the right hand side, the information bit u_k as input and the

This work was mainly done while visiting the University of Notre Dame and was supported by the ETHZ grant TH-16./99-3 and by the Karolus Foundation grant 098/2001-08.

parity bit $x_k^{(1)}$ as output. A typical trellis section of the second RCC has at time index k the state $s_{k-1}^{(2)}$ on the left hand side, the state $s_k^{(2)}$ on the right hand side, the information bit $u_{\pi^{-1}(k)}$ as input and the parity bit $x_k^{(2)}$ as output; the function $\pi(.)$ represents the permutation of the input bits of the first RCC to the input bits of the second RCC. We assume to have no puncturing and so at time k one transmits the three bits u_k, $x_k^{(1)}$ and $x_k^{(2)}$ which results in a designed rate-$1/3$ code. One can use termination of the turbo code as e.g. proposed in [12].

For guaranteeing that the sum-product algorithm works well on a specific factor graph, it is advisable that there are no small cycles so that the factor graphs looks locally tree-like in order that the messages are as independent as possible [10]. To see better what cycles are involved in the factor graph of Fig. 1 we omit all nodes having degree 1 and 2. (Note that in Fig. 1 the function nodes representing the channel have been omitted; but as we assume to transmit over an additive white Gaussian noise (AWGN) channel, no new loops would be created if we appended these channel function nodes and the corresponding edges.) In this way, we obtain a graph which looks like the one in Fig. 2; we call such a graph an interleaver graph (IG). Afterwards, we will see that having no small cycles helps not only the sum-product algorithm but it also helps to avoid low-weight codewords.

III. Low-Weight Codewords

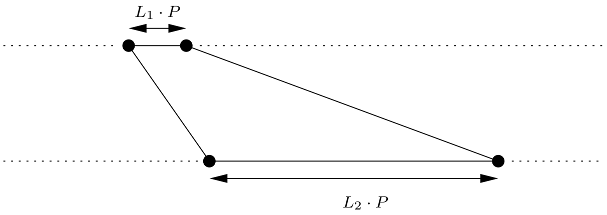

Fig. 3. Pattern in an interleaver graph with 2 input bits being “1” leading to a low-weight codeword if $L_1 + L_2$ is small (P is the period length of the component convolutional codes).

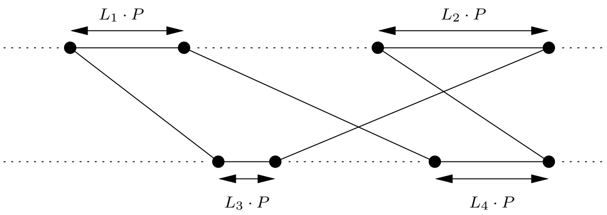

Fig. 4. Pattern in an interleaver graph with 4 input bits being “1” leading to a low-weight codeword if $L_1 + L_2 + L_3 + L_4$ is small (P is the period length of the component convolutional codes).

Assume that both RCCs have memory size ν and that the denominator polynomials (which must not necessarily be the same) are primitive polynomials. Let $P = 2^\nu - 1$ and let us focus on only one of these RCCs. One can show (see e.g. [1]) that if there are two ones at a distance $L \cdot P$ apart at the input ($L \in \mathbb{N}$), only finitely many ones are produced at the output: the weight is not larger than $\alpha \cdot L + \beta$, where α and β are constants depending only on the convolutional encoder. Therefore one should avoid interleaver graphs as depicted in Fig. 3 having connections such that $L_1 + L_2$ is small as this produces codewords of weight not larger than $2 + \alpha_{\mathrm{C1}} L_1 + \alpha_{\mathrm{C2}} L_2 + \beta_{\mathrm{C1}} + \beta_{\mathrm{C2}}$. ($\alpha_{\mathrm{C1}}$, β_{C1}, α_{C2} and β_{C2} are the α’s and β’s of the first and second RCC, resp.) In the same manner, situations as in Fig. 4 should be avoided where $L_1+L_2+L_3+L_4$ is small as this produces codewords of weight not larger than $4+\alpha_{\mathrm{C1}}(L_1+L_2)+\alpha_{\mathrm{C2}}(L_3+L_4)+\beta_{\mathrm{C1}}+\beta_{\mathrm{C2}}$. Of course, this can easily be generalized to longer cycles. Whereas situations as in Fig. 3 are handled by spread (S-random) interleaver designs as in [13] or related design techniques like [14], they cannot handle situations like in Fig. 4.

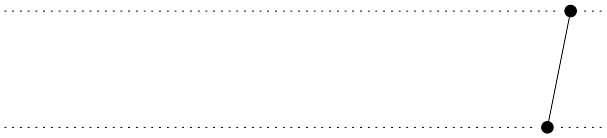

Fig. 5. Pattern with one input bit being “1” leading to a low-weight codeword (see text).

One sees that the codewords of minimum weight are upper bounded by a constant plus a linear function of the shortest length of certain special cycles. In [1] a graph similar to our interleaver graph is derived where one sees these special cycles explicitly. Roughly, as the girth of a graph having degrees larger than two is upper bounded by a constant times the logarithm of the number of vertices, one finally gets the result that the minimum distance can at best only grow proportionally to the logarithm of the interleaver length.

A different situation that should be avoided is depicted in Fig. 5: a low-weight codeword (generated by a single “1” among the input bits) is produced because there is an edge connecting a vertex near the very end of the upper chain and a vertex near the very end of the lower chain.

IV. Cayley and Ramanujan Graphs

In this section we describe the class of graphs which we will use afterwards[1]. Graphs in general consist of vertices and edges whereby each edge connects two vertices; the degree of a vertex is the number of edges incident on it. A directed (undirected) graph has directed (undirected) edges. The diameter of a graph is the maximum distance between any two vertices and the girth is the length of the shortest cycle.

Directed Cayley graphs are defined by specifying a group $\langle \mathcal{G}, \circ \rangle$ and a subset $\mathcal{S} \subseteq \mathcal{G}$ of generators. A directed graph is

[1] Our proposed constructions can start with any graph having large girth and regular degree four. If one is interested in asympotitic existence results, the class of Ramanujan graphs discussed in this section is especially useful.

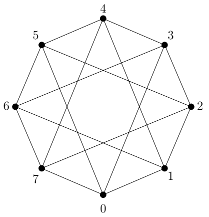

Fig. 6. An undirected Cayley graph with $\langle \mathcal{G}, \circ \rangle = \langle \mathbb{Z}/8\mathbb{Z}, + \rangle$, the integers under addition modulo 8, and the (symmetric) set $\mathcal{S} = \{-3, -1, +1, +3\}$.

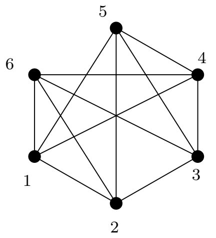

Fig. 7. An example graph with uniform degree 4.

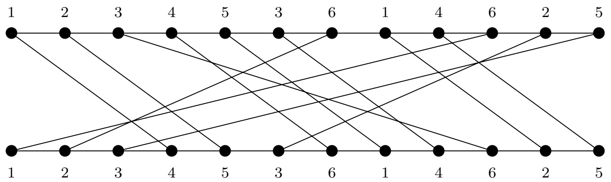

Fig. 8. A first IG derived from the graph in Fig. 7.

then built in the following way: there are $|\mathcal{G}|$ vertices and to each vertex we associate a group element g_i (to make the notation easy we identify the vertex label and the group element); there is a directed edge from vertex g_i to vertex g_j if and only if there is an $s \in \mathcal{S}$ such that $g_j = g_i \circ s$. If the set $\mathcal{S}$ is symmetric, i.e. for each $s \in \mathcal{S}$ we have $s^{-1} \in \mathcal{S}$, the resulting Cayley graph has for each directed edge a directed edge in the opposite direction and one defines an undirected Cayley graph in the following way: one replaces pairs of directed edges by a single undirected edge. Fig. 6 gives an example of an undirected Cayley graph.

Lubotzy, Phillips and Sarnak (LPS) [15] [16] found a special class of graphs which have several execeptional properties, among others they have large girth. Based on other principles, Margulis [17] independently found nearly the same class of graphs. While the reasoning, why these graphs have their special properties, is quite involved, they can quite easily be described as Cayley graphs. To this end, we need several matrix groups. Let $\mathrm{GL}_2(q)$ be the general linear group of all invertible 2×2-matrices with elements from $\mathrm{GF}(q)$ where the group operation is the usual matrix multiplication. The projective general linear group $\mathrm{PGL}_2(q)$ is isomorphic to $\mathrm{GL}_2(q)$ modulo non-zero multiples of the identity matrix: $\mathrm{PGL}_2(q) \cong \mathrm{GL}_2(q)/\left\{a\mathbf{I} \,|\, a \in \mathrm{GF}(q)^*\right\}$. Representatives of the elements of $\mathrm{PGL}_2(q)$ either have a determinant which is a quadratic residue or a quadratic non-residue (i.e. they can/cannot be written as a square in $\mathrm{GF}(q)$). For odd q's the projective special linear group $\mathrm{PSL}_2(q)$ finally is an index-2 subgroup of $\mathrm{PGL}_2(q)$ consisting only of those representatives whose determinant is a quadratic residue.

The class of graphs defined by LPS are Cayley graphs defined by two parameters q and p which must be two different odd primes. Roughly speaking, the size of the resulting graph is a function of q and the graph has uniform degree $p+1$. In [15] [16] they give a construction only for q and p each equal to 1 modulo 4 and they allude to the fact that the construction is actually not restricted to those cases [16]. In this paper we give the construction for any odd primes q and p (afterwards, we will mainly be interested in the case $p = 3$ which results in graphs having regular degree 4.)

If p is a quadratic non-residue modulo q then the underlying group of the Cayley graph is $\mathcal{G} = \mathrm{PGL}_2(q)$ and the number of vertices is $|\mathcal{G}| = q(q^2-1)$, otherwise it is $\mathcal{G} = \mathrm{PSL}_2(q)$ and the number of vertices is $|\mathcal{G}| = q(q^2-1)/2$. The set $\mathcal{S}$ is constructed in the following way. One can show that -1 can always be written as a sum of two squares modulo q, i.e. $-1 = u^2 + v^2 \pmod{q}$ for some $u, v \in \mathbb{Z}$ (if $q = 1 \pmod{4}$ then v can be chosen to be zero). The set $\mathcal{S}$ consists of the $p+1$ matrices of the form[2] $\begin{pmatrix} a+bu+dv & c+du-bv \\ -c+du-bv & a-bu-dv \end{pmatrix}$, where $a, b, c, d \in \mathbb{Z}$ are the solutions of the following diophantine equations. In the case p equals 1 modulo 4 they must fulfill $p = a^2 + b^2 + c^2 + d^2$ (in $\mathbb{Z}$), a odd and greater than zero and b, c, d even. In the case p equals 3 modulo 4 they must fulfill $p = a^2 + b^2 + c^2 + d^2$ (in $\mathbb{Z}$), a odd and greater than zero and b, c odd and d even.

The resulting graphs are called Ramanujan graphs because the second largest eigenvalue of the adjacency matrix is below a certain threshold. Among other special properties their girth is lower bounded by $4\log_p(q) - \log_p(4)$ (which is above the Erdős-Sachs bound [18]) in the case where p is a quadratic non-residue modulo q and by $2\log_p(q)$ in the other case. The diameter is in both cases upper bounded by $2\log_p(n) + 2\log_p(2) + 1$, where n is the number of vertices of the graph.

V. Construction of Interleaver Graphs

As the weight of low-weight codewords is related to the girth of the IG (see Sec. III), we will try to construct IGs whose girth is large. The idea is to derive from the Ramanujan graphs (RGs) descibed in Sec. IV (see also Footnote 1) an IG as e.g. shown in Fig. 2 where we are able to give a lower bound on the girth. A possible way to do this is as follows.

In a first step we start with an RG with arbitrary odd prime $q \neq 3$ and $p = 3$, i.e., the graph has uniform degree $3 + 1 = 4$.

[2] The entries of these matrices are given as elements of $\mathbb{Z}$; they have to be mapped to $\mathrm{GF}(q)$ in the usual way (by the canonical homomorphism) so that one obtains a 2×2-matrix with entries in $\mathrm{GF}(q)$.

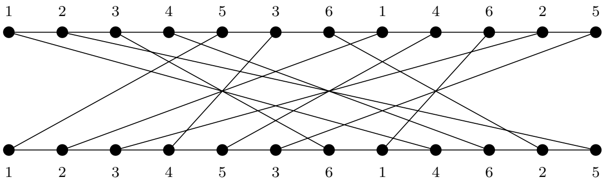

Fig. 9. A second IG derived from the graph in Fig. 7.

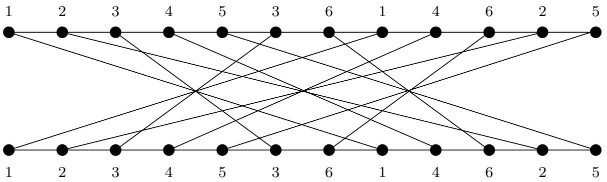

Fig. 10. A third IG derived from the graph in Fig. 7.

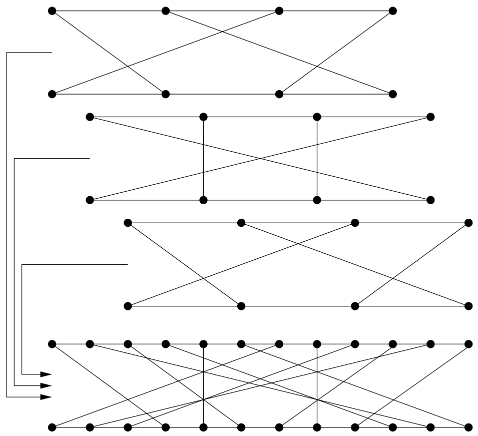

Fig. 11. Out of P small IGs of length 4 a large IG of length 12 is constructed (here $P = 3$).

We take an arbitrary Eulerian walk (EW) in it. (Remember that an EW is a walk in a graph which visits every edge exactly once and starts and ends in the same vertex. Such EWs exist if and only if the degree of all the vertices is even; an EW can be found in time linear in the number of edges.) As the RG has uniform degree 4, each vertex gets visited twice. E.g., Fig. 7 shows a graph with 6 vertices and uniform degree 4 (this is not an RG as defined above, this graph was chosen for demonstration purposes only). A possible EW visits the vertices $1-2-3-4-5-3-6-1-4-6-2-5-(1)$.

In a second step we define the labeling of the upper chain of the IG. (We call the subgraph consisting of the vertices and horizontal edges of the upper part of the IG the upper chain; the subgraph consisting of the vertices and horizontal edges of the lower part is called the lower chain.) We take an upper chain that has twice the number of vertices of the RG and the labeling is done according to the EW chosen before. (Actually, we get a long closed chain, but we cut it at an arbitrary edge.) The lower chain is labeled in the same manner. Continuing the example we started in Fig. 7 we get the upper and lower chain as in Fig. 8.

In a third step we finally define which vertex in the upper chain is connected to which vertex in the lower chain. E.g. for the first vertex of the upper chain with label “1”, this can be done in the following way: determine the right neighbor of the second “1” in the lower chain, which in this case is “4”. The first vertex of the upper chain with label “1” is finally connected to the other occurence of “4” in the lower chain. The other vertices are connected in the same manner and we obtain the IG as shown if Fig. 8.

It is not difficult to see that every path in the IG can be mapped back to a path in the original graph, especially every cycle can be mapped back to a closed path. Therefore, the girth of the interleaver graph is at least as large as in the original graph and we can use the same lower bound on the girth as for the RG. If, as in our case, the original graph is an RG with parameters q and $p = 3$, then the girth grows proportional to the logarithm of q or equivalently proportional to the logarithm of the interleaver length. Finally, also the resulting minimum distance will grow proportionally to the logarithm of the interleaver length.

A different possibility is to connect the first vertex with label “1” in the upper chain to the right neighbor of the second “1” in the lower chain, i.e. to the vertex with label “4” that lies between the vertices with labels “1” and “6”. The other connections are done in a completely analogue manner and we obtain the IG as shown in Fig. 9. Again, the girth can be shown to be at least as good as in the original RG.

Of course, instead to connecting to the right neighbor in the last construction one can also connnect to the left neighbor and still make the same conclusions. In fact, one can also connect the first vertex with label “1” in the upper chain with the second vertex with label “1” in the lower chain and do the other connections analogously (see Fig. 10).

The problem in the above constructions is though in the small constant in front of the logarithm. It is advisable to have a good girth but it is also equally important to avoid certain pattern leading to codewords having low weight as shown in Sec. III. One can resolve this problem with the following construction. For a given P (where P is defined in Sec. III), one constructs in the first step P (small) IGs of the same size using one of the above constructions. In the next step one combines them as shown in Fig. 11 to a new (large) IG. It is not difficult to see that the weight of every low-weight codeword of the form as discussed in Sec. III is a function of the girth of the small IGs and for this girth we can give a lower bound by construction.

Additionally, we can shift the lower chain relatively to the upper chain to avoid unlucky connections at the end of the IG as pointed out at the end of Sec. III (Fig. 5).

As a specific construction we consider the case $q = 5$, $p = 3$. As p is a quadratic non-residue modulo q, the RG has $q(q^2 - 1) = 120$ vertices of degree 4. An EW of such an

RG must be of length $2 \cdot 120 = 240$ and therefore the small IGs have upper and lower chain lengths of 240 (we used different possibilities to get from the EW to the small IG). As component RCCs we choose systematic RCCs of memory size $\nu = 3$ with transfer functions $n_1(D)/d_1(D)$ and $n_2(D)/d_2(D)$, respectively, where $n_1(D) = n_2(D) = 1 + D + D^2 + D^3$, $d_1(D) = 1 + D + D^3$ and $d_2(D) = 1 + D^2 + D^3$; therefore $P = 7$. The large IG of length $7 \cdot 240 = 1680$ consists of 7 copies of the small IG (each of them individually shifted cyclically by a certain offset). As we do no puncturing of the code, we get a designed rate of $1/3$. We use termination of the turbo code, so the final code has length $3 \cdot (1680 + 3) = 5049$ and 1680 information bits. Using the algorithms as proposed in [19] to compute the minimum distance we obtained a minimum distance of 30. This compares favorably with results shown in [20] and [19]: Table 2 of [20] shows that a random interleaver construction of turbo codes with the same interleaver length and memory size $\nu = 3$ ($P = 7$) gives an average minimum distance of roughly 14 and a best minimum distance of 22. Table IV and Fig. 7 of [19] show that random interleaver construction of turbo codes with interleaver length 1280 and memory size $\nu = 4$ ($P = 15$) gives an average minimum distance of 19.7 and a best minimum distance of 30. In Table II of [19] the CCSDS rate-$1/3$ code with interleaver length 1784 and $\nu = 4$ ($P = 15$) is shown to have minimum distance 32. (Note that our construction has only $\nu = 3$ and $P = 7$.)

VI. Conclusions and Outlook

We have considered different ways of deriving IGs that try to avoid certain low-weight codewords. The starting point were graphs where one can give a lower bound on the girth (we especially focused on RGs). We would like to comment different points.

- Hamiltonian cycle: instead of an EW one can take a Hamiltonian cycle. (Remember that a Hamiltonian cycle is a cycle which visits each vertex exactly once; for general graphs it is very hard to find such a cycle if there is one at all.) In this case, instead of starting with a graph with uniform degree four one can of course also start with a graph with uniform degree three.
- Interleaver length: the algorithms presented in [19] work in resonable time for small and medium interleaver lengths. The minimum distance of turbo codes with random interleavers of these sizes can therefore be checked by these algorithms. This is not anymore possible for long interleaver lengths.
- The presented method works especially well for medium sized to long sized interleavers. An important step will be to find good possibilities to derive IGs with more possible choices of interleaver length.
- Description size: if the large IG concists of several copies of the same small IG, one needs only to save the interleaver of the small IG and the offsets (therefore one needs roughly P times less space to save the interleaver).
- In [21] so-called unifilar turbo codes were discussed. It is possible to derive interleavers for such turbo codes in similar ways as proposed in this paper for the derivation of IGs for classical turbo codes.

Acknowledgments

I would like to thank for discussions with my advisor H.-A. Loeliger from ETH Zurich and J. Rosenthal from the University of Notre Dame.

References

[1] M. Breiling, "A Logarithmic Upper Bound on the Minimum Distance of Turbo Codes," *submitted to IEEE Trans. Inform. Theory, available under* `http://www.lnt.de/~breiling`.

[2] G. A. Margulis, "Explicit Constructions of Graphs Without Short Cycles and Low Density Codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, 1982.

[3] G. A. Margulis, "Some New Constructions of Low-Density Parity-Check codes," in *3rd. Intern. Seminar on Information Theory, Convolution Codes and Multi-User Communication*, (Sochi), pp. 275–279, 1987.

[4] J. Rosenthal and P. O. Vontobel, "Constructions of LDPC Codes Using Ramanujan graphs and Ideas from Margulis," in *Proc. of the 38-th Allerton Conference on Communication, Control, and Computing*, (Monticello, Illinois, USA), pp. 248–257, Oct. 4–6 2000.

[5] J. Rosenthal and P. O. Vontobel, "Construction of Regular and Irregular LDPC Codes Using Ramanujan Graphs and Ideas from Margulis," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Washington, D.C., USA), p. 4, June 24–29 2001.

[6] J. Lafferty and D. Rockmore, "Codes and Iterative Decoding on Algebraic Expander Graphs," in *Proc. of ISITA 2000*, (Hawaii, USA), 2000.

[7] O. Y. Takeshita and D. J. Costello, "New Deterministic Interleaver Designs for Turbo Codes," *IEEE Trans. on Inform. Theory*, vol. IT-46, pp. 1988–2006, Sept. 2000.

[8] R. M. Tanner, "Toward an Algebraic Theory for Turbo Codes," in *2nd Intern. Conf. on Turbo Codes and Related Topics*, (Brest, France), pp. 17–25, Sept. 4–7 2000.

[9] L. Bazzi, M. Mahdian, S. Mitter, and D. Spielman, "The Minimum Distance of Turbo-Like Codes," *preprint*, 2002.

[10] N. Wiberg, *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, Sweden, 1996.

[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. on Inform. Theory*, vol. IT–47, no. 2, pp. 498–519, 2001.

[12] D. Divsalar and F. Pollara, "Turbo Codes for Deep-Space Communications," *JPL, TDA Progress Report*, vol. 42-120, Feb. 1995.

[13] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations," *JPL, TDA Progress Report*, vol. 42-122, Aug. 1995.

[14] D. L. Ruyet and H. V. Thien, "Design of Cycle Optimized Interleavers for Turbo Codes," in *2nd Intern. Conf. on Turbo Codes and Related Topics*, (Brest, France), pp. 335–338, Sept. 4–7 2000.

[15] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan Graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[16] P. Sarnak, *Some Applications of Modular Forms*. Cambridge: Cambridge University Press, 1990.

[17] G. A. Margulis, "Explicit Group-Theoretic Constructions of Combinatorial Schemes and Their Applications in the Construction of Expanders and Concentrators," *Problemy Peredachi Informatsii*, vol. 24, no. 1, pp. 51–60, 1988.

[18] P. Erdős and H. Sachs, "Reguläre Graphen gegebener Taillenweite mit minimaler Knotenzahl," *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, vol. 12, pp. 251–257, 1963.

[19] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the Free Distance of Turbo codes and Serially Concatenated Codes with Interleavers: Algorithms and Applications," *IEEE Trans. on Comm.*, vol. COMM-19, pp. 800–812, May 2001, programs available under `http://www.tlc.ee.unian.it./ricerca/turbo/freedistance.html`.

[20] R. Garello, F. Chiaraluce, P. Pierleoni, M. Scaloni, and S. Benedetto, "On Error Floor and Free Distance of Turbo Codes," in *Proc. IEEE Int. Conf. Communications*, vol. 1, (Helsinki, Finland), pp. 45–49, June 11-14 2001.

[21] H.-A. Loeliger, "New Turbo-Like Codes," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Ulm, Germany), p. 109, June 29-July 4 1997.