ON THE CONTENTION-FREE AND SPREAD CHARACTERISTICS OF SERIALLY-PRUNED INTERLEAVERS

Mohammad M. Mansour

ECE Department
American University of Beirut
Beirut, Lebanon
Email: mmansour@aub.edu.lb

ABSTRACT

Serial pruning of turbo interleavers have been proposed in the literature as a simple scheme to provide more flexible codeword lengths. In this paper, we prove two important attributes about serially pruned interleavers. First, we show that serially pruned interleavers inherit the content-free property of their mother interleaver, and hence they remain parallelizable. An example serially-pruned QPP LTE interleaver is parallelized. Second, the minimum spread factor of a serially-pruned interleaver closely matches the spread factor of its mother interleaver for small pruning gaps with minimal impact on BER performance, and degrades gracefully with the pruning length. Simulation results of practical pruned LTE turbo interleavers demonstrate the graceful degradation of spread characteristics and BER performance of serially pruned interleavers.

***Index Terms*—** Contention-free interleavers, serially-pruned interleavers, turbo codes, LTE, QPP interleavers.

1. INTRODUCTION

Interleavers are devices that reshuffle a sequence of symbols according to some permutation [1]. Recently, they have been most commonly employed in the context of turbo codes. A class of computationally efficient interleavers are deterministic interleavers [2,3] with simple address generation expressed in closed-form.

Many practical deterministic interleavers however are limited to a small set of discrete lengths. Pruning is a technique used to support more flexible block lengths [4–6]. Communication standards [7–9] typically vary the block length depending on the input data rate requirements and channel conditions. To support any length β, interleaving is done using a mother interleaver with smallest length $k>\beta$ such that outlier interleaved addresses $\geq \beta$ are excluded. The so-called inliers permutation statistic [10,11] $I(\pi, k; \alpha, \beta)$ enumerates all integers between 0 and $\alpha - 1$ that map to indices less than some integer $0 < \beta < k$ in the permuted sequence.

However, arbitrary pruning alters the spread characteristics of the mother interleaver. It also creates a serial bottleneck since interleaved indices become address-dependent, and hence permuting streaming data in parallel on the fly is no longer practically feasible [12].

A *serially pruned* interleaver (SPI) of size $\alpha < k$ and pruning length $\beta < k$, with $\alpha < \beta$, is defined by $\mathring{\pi} : \mathcal{D} \rightarrow \mathcal{R}$, $x \mapsto y = \mathring{\pi}(x) = \pi(p(x))$, where $|\mathcal{D}| = |\mathcal{R}| = \alpha$, such that: 1) $\mathring{\pi}(x) < \beta$, and 2) $p(x) \triangleq x + \Delta_x$ is the pruning function where Δ_x is the pruning gap of x defined to be the minimum $\Delta \geq 0$ such that $I(\pi, k; x+\Delta, \beta) = x$ (i.e., for $j = 0, \cdots, x+\Delta_x-1$, $\pi(j) < \beta$ is satisfied exactly x times). The domain and range of $\mathring{\pi}$ are $\mathcal{D} = [\alpha]$ and $\mathcal{R} = \pi(p([\alpha]))$. If this gap can be efficiently computed, then pruned interleaving can be parallelized by windowing using the minimal inliers and parallel pruning algorithms in [10]. Also this gap is used to characterize the minimum spread of a SPI as shown later.

Serial pruning is valuable in turbo coding applications because of the flexibility it gives in tuning the codeword length. Typically, in a communication system employing adaptive modulation and coding, only a small set of discrete codeword lengths k are supported. Bits are either punctured or filled in to match the nearest supported length. For a pruned interleaver $\mathring{\pi}$ of length β to be useful, it is desirable to have the following characteristics: 1) It does not require extra storage memory to store the pruned indices, 2) pruning preserves the contention-free property [13, 14] of its mother interleaver (if present), and 3) its spread factor [15] degrades gracefully with the number of pruned indices $g \triangleq k - \beta$, and hence the impact on BER performance is limited.

Obviously serial pruning satisfies the first property. The contribution of this paper is that we prove that serial pruning also satisfies the other two properties. In particular, we show that serially pruned interleavers inherit the content-free property of their mother interleaver, and hence they remain parallelizable. This is addressed in Section 2. Furthermore, by deriving a lower bound on the minimum spread of a SPI, we show that spread factor still matches closely the spread factor of its mother interleaver for small g. This is treated in Section 3 together with simulation plots that confirm this result. The reader is referred to our earlier work for more details on pruned interleavers, their characteristics, and associated parallel pruning algorithms and architectures in [10, 11].

2. CONTENTION FREE PROPERTY

A permutation of length $k = W{\cdot}M$ in general is said to be contention-free [13] with degree $M = 2^m$, if an array of k data elements stored in one set of M *read* memory banks, each of size $W = 2^w$, can be permuted and written into another set of M *write* memory banks, such that at each step, M data elements are read in parallel from the M read banks and written in parallel into the M write banks without reading or writing more than one element from/to any bank (see Fig. 1a). Data is stored sequentially in the read banks such that linear address $i = j + tW$ corresponds to location j in bank $t = \lfloor i/W \rfloor$, where $0 \leq j < W$ and $0 \leq t < M$. To permute any set of M data entries at linear addresses $j, j+W, \cdots, j+(M-1)W$ in parallel, the contention-free property stipulates that

$$B\left(\pi(j+tW); W, M\right) \neq B\left(\pi(j+vW); W, M\right), \qquad (1)$$

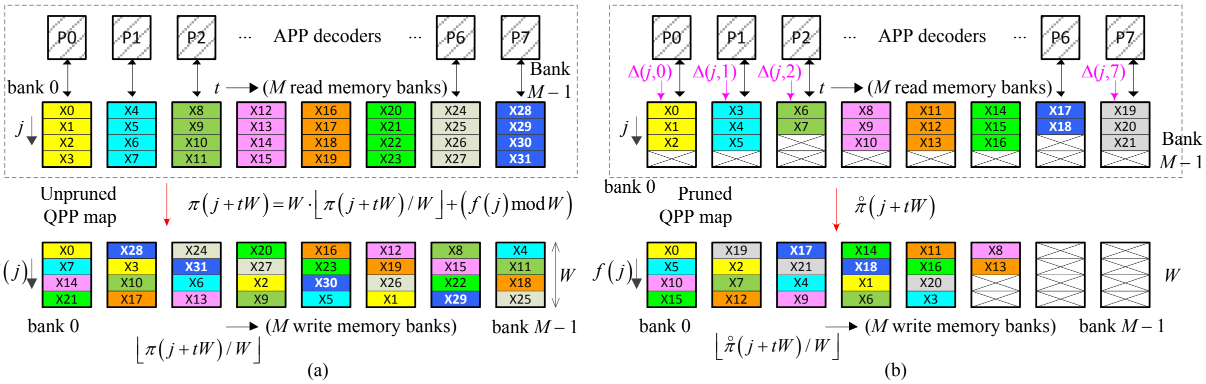

Fig. 1: 8-way parallel contention-free mapping for (a) an unpruned, and (b) a pruned QPP map $\pi(j) = f(j) \bmod k$ with $f(j) = 7j + 16j^2$, $k = 32$, $\alpha = \beta = 22$, $W = 4$, and $M = 8$.

for all $0 \leq j < W$ and $0 \leq t \neq v < M$, where the bank addressing function $B()$ is defined to be either $B(i; W, M) \triangleq \lfloor \frac{i}{W} \rfloor$ or $B(i; W, M) \triangleq x \bmod M$. This is a more general condition than [13], and effectively uses either the m most or least significant bits (MSBs/LSBs) of $\pi(j+tW)$ as a permuted bank address.

Figure 1a illustrates the contention-free property of a quadratic permutation polynomial (QPP) $\pi(j) = 7j + 16j^2 \pmod k$ for $k = 32, W = 4$, and $M = 8$, in the context of turbo decoding [13, 14]. Eight constituent a-posteriori probability (APP) decoders operate in parallel to decode a codeword by reading log-likelihood rations (LLRs) values in parallel from $M = 8$ banks each of size $W = 4$, and writing QPP-permuted LLR values in parallel to the write banks without contention.

It is easy to show that the contention-free property extends beyond QPP permutations of degree 2 to the class of polynomial-based permutations [16, 17] of the form $\pi(j) = f(j) \pmod k$, where $f(j) = \sum_{i=0}^{d} a_i j^i$ is a degree-d polynomial with appropriately chosen coefficients a_i over the ring of integers modulo k. To prove this, we select the bank addressing function to be $B(i; W, M) \triangleq \lfloor \frac{i}{W} \rfloor$, so that the m MSBs in the address designate a permuted bank address, and prove that (1) holds. We can write $f(j)$ as

$$f(j) = k \left\lfloor \frac{f(j)}{k} \right\rfloor + \pi(j)$$

Then for any W that divides k, we have $f(j) = \pi(j) \pmod W$. Therefore,

$$\begin{aligned} \pi(j) &= W \left\lfloor \frac{\pi(j)}{W} \right\rfloor + (\pi(j) \bmod W) \\ &= W \left\lfloor \frac{\pi(j)}{W} \right\rfloor + (f(j) \bmod W) \end{aligned} \tag{2}$$

Now since $f(j)$ is a polynomial function in j, we have for any t

$$f(j + tW) = f(j) \pmod W \tag{3}$$

For distinct windows t and u, we have $\pi(j + tW) \neq \pi(j + uW)$ since π is a permutation, which implies using (2) and (3) that

$$\left\lfloor \frac{\pi(j + tW)}{W} \right\rfloor \neq \left\lfloor \frac{\pi(j + uW)}{W} \right\rfloor$$

Furthermore, the bit-reversal permutation (BRP) [10] $\pi_n(j)$ on n bits [10] is contention-free as well for any $k = 2^n, M = 2^m, W = 2^w$, where $n = m + w$ and $m < n$. The proof is based on the following property of bit-reversal permutations [10]: $\pi_n(j + tW) = M \cdot \pi_w(j) + \pi_m(t)$, where $\pi_w(j)$ and $\pi_m(j)$ are bit-reversal permutations on w and m bits, respectively. Then, for any pair of distinct windows t, u, we have $\pi_n(j + tW) \neq \pi_n(j + uW)$, from which we obtain

$$M \cdot \pi_w(j) + \pi_m(t) \neq M \cdot \pi_w(j) + \pi_m(u) \pmod M,$$

for $j = 0, 1 \cdots, W - 1$. Hence, $\pi_m(t) \neq \pi_m(u) \pmod M$, and thus the m LSBs designate a permuted bank address.

Next, applying an arbitrary pruning to a permutation to shorten its length to $\beta < k$ does not in general preserve its contention-free property. However, we show next that serial pruning does indeed preserve this property, and a contention-free pruned permuter can be designed as shown in theorem 1 below. First, it is important to note that the serial-pruning map $p(i) = i + \Delta_i$ itself is contention free. To show this, take two addresses $i_1 = j + t_1 W$ and $i_2 = j + t_2 W$ that correspond to banks t_1 and $t_2 > t_1$. Then $\lfloor (j + t_1 W + \Delta_{i_1})/W \rfloor \neq \lfloor (j + t_2 W + \Delta_{i_2})/W \rfloor$ for any $0 \leq j < W$ since $p(\cdot)$ is monotonically increasing and hence $\Delta_{i_2} \geq \Delta_{i_1}$.

Theorem 1. *Any serially-pruned, contention-free permutation (interleaver) remains contention free after pruning.*

Proof. One scenario is to insert zero filler bits in the pruned positions while storing the data sequentially in memory across the banks. This requires comparing $\pi(j)$ with β serially for every j before writing to memory. Hence the contention-free property applies for the pruned interleaver across all the banks if the mother interleaver is contention-free.

Another scenario is to store the data across the banks without filler bits as shown in Fig. 1b. To interleave properly, we need to keep track of the inliers that fall within each window. First, since the number of inliers up to window t is $\Delta_{(t+1)\cdot W} = I(\pi, k; (t + 1) \cdot W, \beta)$, data located between address $\Delta_{t \cdot W}$ and $\Delta_{(t+1)\cdot W} - 1$ are stored sequentially in bank t. We know that addresses $j, j + W, \cdots, j + (M-1)W$ map to distinct windows under π. Address j in window t, which might be pruned, actually corresponds to the unpruned address $j + tW - \Delta(j, t)$, where $\Delta(j, t)$ is defined as:

$$\Delta(j,t) = \begin{cases} \Delta_{t \cdot W}, & \text{if } j = 0; \\ \Delta(j-1, t), & \text{if } j > 0,\ \pi_n(j + tW) < \beta; \\ \Delta(j-1, t) + 1, & \text{if } j > 0,\ \pi_n(j + tW) \geq \beta. \end{cases} \tag{4}$$

with initial condition $\Delta_0 = 0$. Then, for $0 \leq j < W$ and $0 \leq t \neq v \leq M$, we have

$$B\left(\overset{\circ}{\pi}(j+tW-\Delta(j,t));W,M\right)=B(\pi(j+tW);W,M)$$
$$\neq B(\pi(j+vW);W,M)=B\left(\overset{\circ}{\pi}(j+vW-\Delta(j,v));W,M\right)$$

Hence a serially-pruned interleaver is contention-free when the banks are accessed sequentially using a counter from $j=0,1,\cdots,W-1$, if the mother interleaver is contention free. $\square$

In Fig. 1b, the theorem is applied to the QPP of Fig. 1a after serially pruning it to length $\beta=22$ in order to permute its entries in parallel without contention when accessing the memory banks. Eight APP decoders are still included in the figure, where it is assumed that the hardware can be reconfigured on the fly to decode codeword of any pruned length $k/2<\beta\leq k$. Pruned entries in the memory banks are marked as ⊠. Each read memory bank t is initialized with the appropriate $\Delta(j,t)$ using (4), and accessed by a counter j that runs from 0 to $W-1$. When reading from bank t at step j, the actual address corresponds to $j+tW-\Delta(j,t)$. If $\pi(j+tW)<\beta$, the read is successful. Otherwise, the location is pruned, reading from bank t is stalled and $\Delta(j,t)$ is incremented. At step j, at most 6 APP decoders operate and perform 6 parallel reads from the memory banks. The generated LLR values from the decoders are written in parallel in pruned QPP permuted order in the write memory banks in 4 steps.

The pruning gaps in (4) can be computed efficiently using the Minimal Inliers algorithm in [10] together with any scheme to enumerate the inliers depending on the permutation at hand. Efficient schemes to enumerate inliers of LPPs, BRPs, and 2D interleavers have been treated in our earlier work (see [10, 11]). The implications are that serially-pruned contention-free interleavers are parallelizable at a low implementation cost using the proposed schemes. When coupled with windowing techniques to parallelize the constituent APP decoders, a turbo decoder can then be efficiently parallelized to meet throughput requirements in 4G wireless standards and beyond.

It is worth noting that pruning can also be employed to design more efficient radix-2 FFTs by eliminating redundant or vacuous computations when the input vector has many zeros and/or when the required outputs may be very sparse compared to the transform length. Assume in-place FFT computations using a set of butterflies that compute the final outputs in a set of memory banks in bit-reversed order. We showed above the BRP is contention-free. Hence the last FFT stage for bit-reversal re-ordering can be parallelized. With pruning, the BRP stage can still be parallelized. In fact, a similar memory architecture to that in Fig. 1 can be employed to this effect. The details are omitted due to lack of space.

3. BOUND ON MINIMUM SPREAD

The spread factor of an interleaver is a popular measure of merit for turbo codes [15]. In this section, we consider the impact of serial pruning on the spread characteristics of an interleaver. Let the spread measures of a mother interleaver π and a serially pruned interleaver $\overset{\circ}{\pi}$ associated with two indices i,j be denoted by $S(i,j)=|\pi(i)-\pi(j)|+|i-j|$ and $S_p(i,j)=|\overset{\circ}{\pi}(i)-\overset{\circ}{\pi}(j)|+|i-j|=|\pi(p(i))-\pi(p(j))|+|i-j|$, respectively. The *minimum* spreads of π and $\overset{\circ}{\pi}$ are defined as $S_{\min}\triangleq\min_{i,j<k}S(i,j)$ and $S_{p,\min}\triangleq\min_{i,j<\beta}S_p(i,j)$, $i\neq j$. We prove in the following theorem that $S_{p,\min}$ remains close to $S_{\min}$ when the number of pruned indices $g\triangleq k-\beta$ is small.

Theorem 2. *The minimum spread of a serially-pruned interleaver of length β is at least*

$$S_{p,min}\geq\frac{S_{min}}{(1+\gamma+g/k)^t}\tag{5}$$

where γ is a small positive constant and

$$t=-\log(1-\gamma-g/k)/\log(1+\gamma+g/k).$$

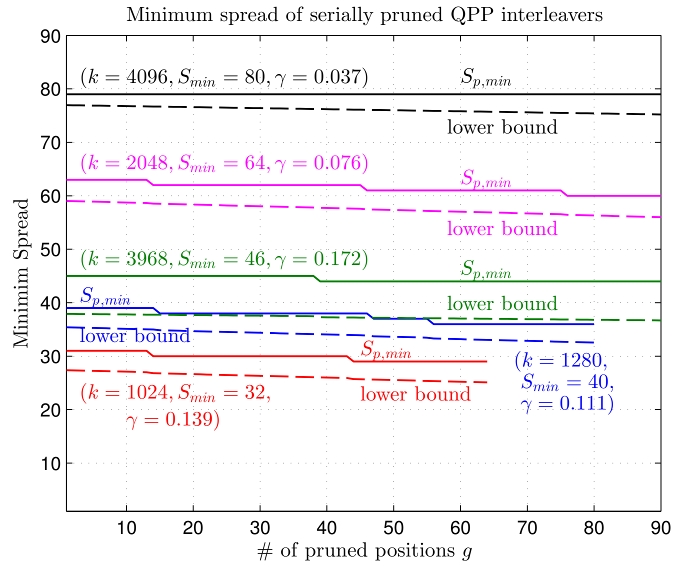

Fig. 2: Minimum spread of pruned QPP interleavers in LTE.

The proof relies on the fact that $S_{p,\min}+|p(i_0)-p(j_0)|-|i_0-j_0|\geq S_{\min}$, where i_0,j_0 are such that $|\pi(p(i_0))-\pi(p(j_0))|+|i_0-j_0|=S_{p,\min}$. The difference $D\triangleq|p(i_0)-p(j_0)|-|i_0-j_0|$ is upper bounded as $D\leq p(j_0)-j_0$, assuming $j_0>i_0$, since $p(\cdot)$ is a monotonically increasing function. Since i_0,j_0 cannot be separated by more than $S_{p,\min}-1$ positions, we need to find the maximum of $p(j_0)-j_0$ when $j_0=S_{p,\min}$. This is equivalent to finding the maximum expansion of an interval of length $S_{p,\min}$ such that it contains *at least* $S_{p,\min}$ inliers. Using the Minimal Inliers algorithm from [10], together with the fact that the minimum number of $\alpha\beta$-inliers $I(\pi,k;\alpha,\beta)$ and the maximum number of $\alpha\beta$-outliers $O(\pi,k;\alpha,\beta)$ in an interval with $\alpha<\beta$, are bounded as

$$\alpha\beta/k-c_1<I(\pi,k;\alpha,\beta)$$
$$O(\pi,k;\alpha,\beta)<\alpha-\alpha\beta/k+c_1$$

where c_1 is a constant, this expansion leads to finding the minimum $t\geq0$ that satisfies $S_{p,\min}(1+\gamma+g/k)^t(1-\gamma-g/k)\geq S_{p,\min}$, from which (5) follows. For example, the QPP interleaver $\pi(j)=63j+128j^2 \pmod{2048}$ has $S_{\min}=64$ and $\gamma=0.076$. If $g=20$ positions are pruned, then $S_{p,\min}\geq58$. In fact, the actual $S_{p,\min}$ is 62. $\square$

Figure 2 plots the minimum spread of serially pruned QPP interleavers as a function of g, for several mother QPP interleavers. The lower bound in (5) is plotted as well. The length k, minimum spread $S_{\min}$ and constant γ of the mother interleavers are shown in brackets. As shown, $S_{p,\min}$ of the pruned interleavers remains very close to $S_{\min}$ when up to $g=2S_{\min}$ indices are pruned, and the lower bounds predicted by (5) are rather tight.

To assess the impact of serial pruning on error-correction performance, the BER of 3GPP LTE turbo codes employing serially pruned QPP interleavers were simulated over an AWGN channel, assuming BPSK modulation and log-MAP decoding with up to 6 de-

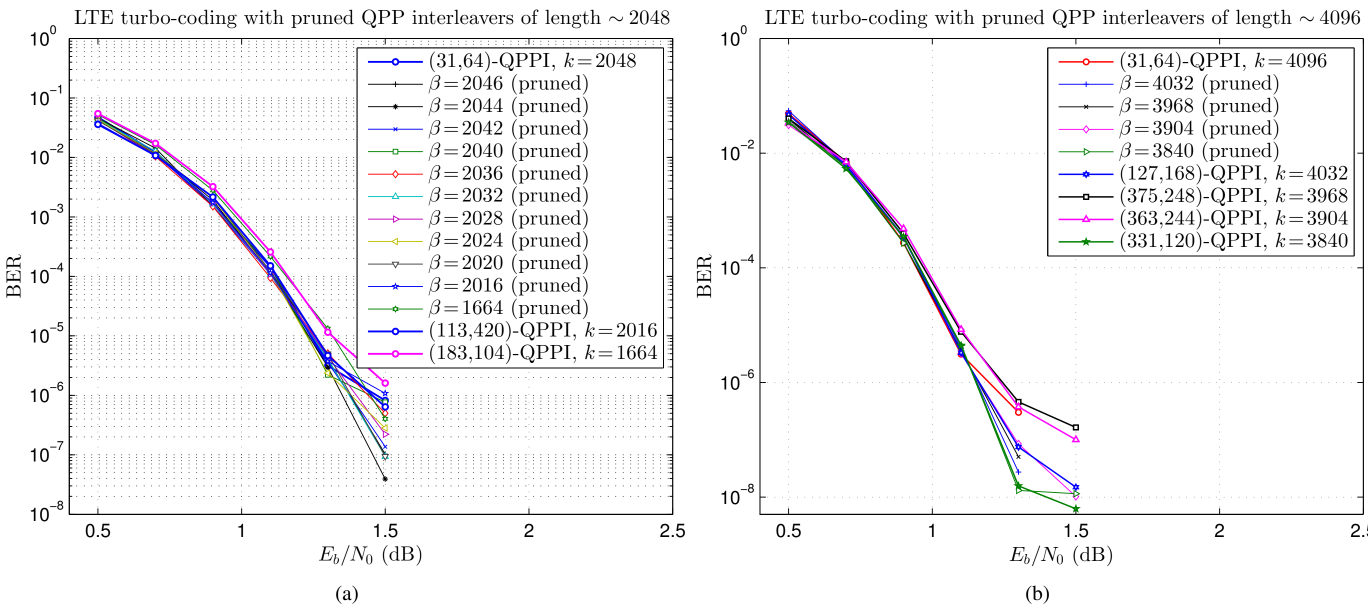

Fig. 3: BER of LTE turbo codes with pruned QPP interleavers: (a) $k=2048$ and (b) $k=4096$.

coding iterations. 500,000 frames were simulated. Figure 3a shows the results using the LTE QPP mother interleaver $\pi(j)=31j+64j^2 \pmod{2048}$ with $k=2048$ and $S_{\min}=64$. Eleven interleavers were derived from this mother interleaver whose lengths are indicated in the figure. Also shown for comparison are results for two other QPP interleavers of lengths 2016 and 1664 that are supported in LTE (the other 9 lengths are not supported). In almost all cases, the pruned interleavers perform very close to, or even outperform, the QPP interleavers. Figure 3b shows the results for a length 4096 QPP mother interleaver with $S_{\min}=80$, and 4 serially pruned interleavers with the indicated lengths. All 4 pruned interleavers outperform the mother interleaver, except one which shows an error floor at 1.5dB. Also compared are four QPP interleavers of similar lengths that are supported in LTE. Again the pruned interleavers outperform or closely match these QPP interleavers.

4. REFERENCES

[1] J. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inf. Theory*, vol. 16, no. 3, pp. 338–345, May 1970.

[2] R. Garello, G. Montorsi, S. Benedetto, and G. Cancellieri, "Interleaver properties and their applications to the trellis complexity analysis of turbo codes," *IEEE Trans. Commun.*, vol. 49, no. 5, pp. 793–807, May 2001.

[3] O. Y. Takeshita and D. J. Costello, Jr., "New deterministic interleaver designs for turbo codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 1988–2006, Sep. 2000.

[4] M. Eroz and A. R. Hammons, Jr., "On the design of prunable interleavers for turbo codes," in *Proc. IEEE Veh. Tech. Conf. (VTC)*, Houston, Texas, USA, May 1999, vol. 2, pp. 1669–1673.

[5] M. Ferrari, F. Scalise, and S. Bellini, "Prunable S-random interleavers," in *Proc. IEEE Conf. on Commun. (ICC)*, New York City, New York, USA, Apr. 2002, vol. 3, pp. 1711–1715.

[6] L. Dinoi and S. Benedetto, "Design of fast-prunable S-random interleavers," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2540–2548, Sep. 2005.

[7] "Evolved universal terrestrial radio access (E-UTRA): Multiplexing and channel coding," Tech. Rep. 3GPP TS 36.212, 3rd Generation Partnership Project (3GPP), Sep. 2008.

[8] "IEEE standard for local and metropolitan area networks — part 20: Air interface for mobile broadband wireless access systems supporting vehicular mobility," Tech. Rep. 802.20, IEEE, Piscataway, NJ, 2008.

[9] "IEEE standard for local and metropolitan area networks — part 16: Air interface for broadband wireless access systems," Tech. Rep. 802.16, IEEE, Piscataway, NJ, 2009.

[10] M. M. Mansour, "A recursive algorithm for pruned bit-reversal permutations," in *Proc. IEEE Conf. on Acoustics, Speech, and Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012.

[11] M. M. Mansour, "Parallel lookahead algorithms for pruned interleavers," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3188–3194, Nov. 2009.

[12] A. Parsons, "The symmetric group in data permutation, with applications to high-bandwidth pipelined FFT architectures," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 477–480, Jun. 2009.

[13] A. Nimbalker, T. K. Blankenship, B. Classon, T. E. Fuja, and D. J. Costello, Jr., "Contention-free interleavers for high-throughput turbo decoding," *IEEE Trans. Commun.*, vol. 56, no. 8, pp. 1258–1267, Aug. 2008.

[14] O. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1249–1253, Mar. 2006.

[15] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," Tech. Rep., JPL TDA Progress Report 42-122, Aug. 1995.

[16] J. Sun and O. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 101–119, Jan. 2005.

[17] O.Y. Takeshita, "Permutation polynomial interleavers: An algebraic-geometric perspective," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2116–2132, Jun. 2007.