This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2017.2718523, IEEE Communications Letters

An Imprecise Stopping Criterion based on in-Between Layers Partial Syndromes

D. Declercq, V. Savin, O. Boncalo and F. Ghaffari

***Abstract*—In this letter we address the issue of early stopping criterion for layered LDPC decoders, aiming at more safeness with low hardware cost and minimum latency. We introduce a new on-the-fly measure in the decoder, called in-between layers partial syndrome (IBL-PS), and define a family of stopping criteria, with different tradeoffs between complexity, latency and performance. Numerical results show that our stopping criteria surpass existing solutions, and can be as safe as the full-syndrome detection, down to frame error rates (FER) as low as FER=10^{-8}.**

***Index Terms*—layered LDPC decoder, early termination, in-between layers partial syndrome.**

I. Introduction

Low-Density Parity-Check (LDPC) codes have attracted much attention in the past several years [1] due to their excellent performance under iterative decoding. Iterative decoding of LDPC codes suffer from a high latency issue, since the decoding iterations need to be performed sequentially. In order to solve this latency issue, it is important to limit the number of decoding iterations needed to correct the channel errors. Since the average number of iterations is typically much smaller than the maximum number of iterations, stopping the decoder as early as possible may lead to significant improvement in terms of both average throughput and energy consumption [2], [3]. The classical approach to declare convergence to a codeword is to compute the syndrome of the LDPC code and to verify that it is equal to zero. However, this straightforward implementation of the stopping criterion impacts an increase of the hardware (HW) cost, or a degradation of the decoding latency.

For the class of structured quasi-cyclic LDPC codes (QC-LDPC) [4], specific stopping criteria have been proposed, benefiting from the organization of the decoding flow of the layered architecture [5]. Of special interest are the low cost early stopping criteria which can be easily integrated in the layered decoding flow, *i.e.* without halting the decoding process, referred to hereafter as *on-the-fly* stopping criteria.

Multiple strategies have been employed for performing early termination for QC-LDPC codes, which can be classified in two main types: *(i)* parity-check (PC) computation based [6], [7], and *(ii)* a posteriori log-likelihood ratio (AP-LLR) based [2], [8], [9], [3]. The lowest complexity approaches usually come at the price of some performance degradation in error correction, in which case the stopping criterion is said *unsafe* or *imprecise* [7]. Unsafe criteria lead to residual errors on the hard-decision bits, which are qualified as *undetected errors*. The safe solutions usually suffer from the drawback of a slower convergence rate, leading to a loss in average throughput and energy consumption.

In this letter, we propose a stopping criterion based on a new measure, called in-between layers partial syndrome (IBL-PS), which consists of checking the PCs of two consecutive layers of QC-LDPC codes. This measure has not been discussed previously in the literature, and is especially interesting for layered decoding, since it can be computed on-the-fly. We propose a family of imprecise stopping criteria, based on the computation and use of one to several IBL-PSs, with different tradeoffs between latency and performance.

The rest of the letter is organized as follows. Section II introduces the concept of IBL-PS, and describes briefly its advantages. In section III, we give a characterization of the decoding situations for which the IBL-PS stopping criterion is unsafe. Section IV shows the performance results of our approach compared with other stopping criteria, and the conclusion is drawn in section V.

II. Concept of in-Between Layers Partial Syndrome

A. Definitions

Quasi-Cyclic LDPC codes are build from blocks of circulant matrices [4]. A QC-LDPC code is defined by a base matrix B, with integer entries $b_{i,j} \geq 0$, $\forall i = 1 \dots M_b$, $\forall j = 1 \dots N_b$. The matrix B is also referred to as *protograph* [11]. The parity-check matrix H of the LDPC code is obtained by expanding the base matrix by a factor L. Each nonzero entry of B is replaced by $b_{i,j}$ circulant matrices of size $L \times L$. In this paper, we discuss only the case of protograph LDPC of type-I, *i.e.* when $b_{i,j} \in \{0, 1\}$. The parity-check matrix of a QC-LDPC code, of size $(M, N) = (M_b\, L, N_b\, L)$, is organized in *layers*. A layer is defined by the set of L consecutive parity-checks, corresponding to one row of circulants.

LDPC codes are decoded by message-passing (MP) decoders that exchange messages between parity-checks and codeword bits. Accordingly, an LDPC decoder comprises two types of processing units, namely check node units (CNUs) and variable node units (VNUs). The specific structure of QC-LDPC codes makes them suitable for HW implementations, when they are decoded using layered scheduling [5]. In layered

D. Declercq and F. Ghaffari are with ETIS, ENSEA/University of Cergy-Pontoise/CNRS, 95014 Cergy-Pontoise, France (declercq@ensea.fr). O. Boncalo is with the University Politehnica Timisoara, Computer Engineering Department, Timisoara, Romania, (boncalo@cs.upt.ro). V. Savin is with the CEA-LETI, MINATEC Campus, Grenoble, France, (valentin.savin@cea.fr).

This work has been supported by the Franco-Romanian (ANR-UEFISCDI) Joint Research Programme Blanc-2013, project DIAMOND.

decoders, parity-checks within one layer are processed in parallel, by instantiating in hardware the corresponding number of CNUs. Messages computed by the CNUs are immediately passed to the corresponding variable nodes, which are then updated before the next layer is processed. At the end of each layer processing, during the VNU step, the AP-LLR messages are updated, from which the hard decision vector is computed, $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)^T \in \{0,1\}^N$. The syndrome vector is defined as $\boldsymbol{s} = H \times \hat{\boldsymbol{x}}$, and when $\boldsymbol{s} = \mathbf{0}$, all the parity-checks are satisfied and the vector $\hat{\boldsymbol{x}}$ is a codeword of H.

Using the submatrix H_i, $\forall i = 1 \ldots M_b$, composed of the L PC equations of a single layer, it is also possible to define a *partial syndrome* vector $\boldsymbol{s}_i = H_i \times \hat{\boldsymbol{x}}$, corresponding to the satisfiability of all the PCs within layer H_i. In the rest of the letter, we will refer to H_i as the i-th layer or the submatrix deduced from it, indifferently. We define similarly the $(2\,L, N)$ submatrix corresponding to the concatenation of two consecutive layers $H_{i,i+1} = [H_i\,;\,H_{i+1}]$, $\forall i = 1 \ldots M_b$, where the indices are taken modulo-M_b. We define the in-between layers partial syndrome (IBL-PS) as follows.

Definition 1: Let $\hat{\boldsymbol{x}}_i = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$ be the hard decision vector, computed after processing of layer H_i and before processing of layer H_{i+1}. The IBL-PS is defined as

$$\boldsymbol{s}_{i,i+1} = H_{i,i+1} \times \hat{\boldsymbol{x}}_i \tag{1}$$

The vector $\boldsymbol{s}_{i,i+1}$ checks the parity of two consecutive layers, using the hard-decision vector available *between* the two layers processing. Note that during one decoding iteration, M_b IBL-PSs can be computed: $\boldsymbol{s}_{1,2}$, $\boldsymbol{s}_{2,3}$, $\ldots$, $\boldsymbol{s}_{M_b,1}$.

For simplicity, we assume that the two layers are consecutive. However, the definition of IBL-PS also applies for any pair of layers (H_i, H_j), with $1 \leq i \neq j \leq M_b$, a long as the two layers H_i and H_j are instantiated consecutively during the layered decoder, through a scheduling controler.

B. On-the-fly Computation of IBL-PS in a Layered Decoder

In [6], the authors have proposed to use the partial syndrome $\boldsymbol{s}_i$ as a measure to define a stopping criterion. They called $\boldsymbol{s}_i$ *on-the-fly syndrome* (OTF syndrome) since its computation does not require to stop the decoder, and is computed with the most recent hard-decision values. From the hardware efficiency point-of-view, this is a real advantage, since computing the sequence of OTF syndromes $\boldsymbol{s}_i$, $\forall i = 1 \ldots M_b$ does not degrade the decoding latency, as opposed to computing the whole syndrome of H.

The IBL-PS measure can also be computed on-the-fly, from the AP-LLR values. At each layer processing, the AP-LLR values are first read from the memory and directed to the appropriate processing units. When layer processing completes, the AP-LLR memory is updated with the newly computed values. Hence, the proposed IBL-PS $\boldsymbol{s}_{i,i+1}$ can be computed by evaluating the parity-checks within layer H_i when data is written back to the AP-LLR memory (*i.e.*, after current layer processing), and those within layer H_{i+1} when data is read from the AP-LLR memory (*i.e.*, before next layer processing). In this way, we ensure that the parity-checks within both layers are verified on the same hard decision vector.

The definition of IBL-PS can be further extended to *generalized layers* (GL). A GL is composed of a number of consecutive rows in B that do not overlap with each other, meaning that any column of B has at most one non-negative entry within these rows.

C. Using the IBL-PS as Imprecise Stopping Criterion

The authors of [6] proposed to stop the decoder when all the OTF syndromes for one iteration, $\boldsymbol{s}_i$, $\forall i = 1 \ldots M_b$ are satisfied. Although of very low complexity, this stopping criterion is unsafe and leads to a large performance degradation for most LDPC codes. In our work, we propose to use the new on-the-fly IBL-PS measures to define a family of stopping criteria, parametrized by $\theta \geq 1$, as follows:

Stopping Criterion 1: The layered decoding stops when θ consecutive IBL-PSs are satisfied, *i.e.* if for some i,

$$\text{IBL-PS}(\theta) \;\Leftrightarrow\; \boldsymbol{s}_{i,i+1} = \boldsymbol{s}_{i+1,i+2} = \ldots = \boldsymbol{s}_{i+\theta-1,i+\theta} = \mathbf{0} \tag{2}$$

Note that if $\theta > M_b$, the IBL-PSs span more than one decoding iteration. Within this family of stopping criteria, the minimum parameter value is $\theta = 1$, and corresponds to the lowest latency case, where only one IBL-PS is computed. It is expected that increasing the value of θ will render the stopping criterion safer and safer.

The hardware resource needed to implement the IBL-PS(θ) criterion is limited as it requires only simple parity check operations that are implemented using XOR gates. Implementation results for FPGA technology have shown that the overhead required to implement the IBL-PS stopping criterion is similar to the overhead for OTF, and represents less than 2% of the global layered architecture implementation cost.

We obtain therefore a family of low-complexity criteria with different tradeoffs between latency (average number of iterations) and performance (safeness).

III. Weaknesses of the IBL-PS(1) Stopping Criterion

Almost all reduced complexity stopping criteria are unsafe, in the sense that there are decoding situations when the decoder is stopped, while the output of the hard decision is not a valid codeword. This leads to undetected errors, and even if in some applications an external code is used to detect those situations, this is a major drawback of low-complexity stopping criteria. Under additional constraints, a low-complexity stopping criterion can be turned into a safe one, but at the cost of extra hardware resource and/or extra latency [3].

For the IBL-PS(1) stopping criterion, early decision resulting in undetected errors is due to specific structures of the Tanner graph of the LDPC code. Those structures can be categorized into different types of graph topologies, which are cycles or trapping sets [1]. A vector that satisfies the partial syndrome $\boldsymbol{s}_{i,i+1}$ will be referred to as a codeword of $H_{i,i+1}$, while a codeword of H is a vector that satisfies the full syndrome $\boldsymbol{s}$. An undetected error of IBL-PS(1) will happen when $\hat{\boldsymbol{x}}$ is a codeword of $H_{i,i+1}$, but not a codeword for H.

	$(M_b, N_b) = (3, 6)$		$L = 128$	
	10-cycles	12 cycles	14-cycles	16-cycles
$H_{1,2}$		2 688		49 536
$H_{2,3}$		5 248		46 848
$H_{3,1}$		4 224		47 488
H	9 600	93 312	756 224	6 299 008

TABLE I: Statistics of the LDPC codes designed specifically for the IBL-PS(1) stopping criterion.

Codewords of $H_{i,i+1}$ are quite simple to characterize in the case of two *full layers*, *i.e.*, when $b_{i,j} = 1$ and $b_{i+1,j} = 1$, $\forall j$. In this case, it is known that only cycles belonging to $H_{i,i+1}$ or combinations of two or more of these cycles give rise to codewords of $H_{i,i+1}$ [12]. It is also known that the girth (length of a shortest cycle) of $H_{i,i+1}$ is at most 12 [4], and hence its minimum distance is less than or equal to 6.

As a consequence, LDPC codes well adapted to the IBL-PS(1) should have the property that any two consecutive layers have the maximum possible girth $g = 12$, with the lowest multiplicity of those 12-cycles. Aside from the cycles themselves, Figure 1 depicts the smallest trapping sets with weights $(8, 9, 10)$ bits, that can stop the decoder under IBL-PS(1).

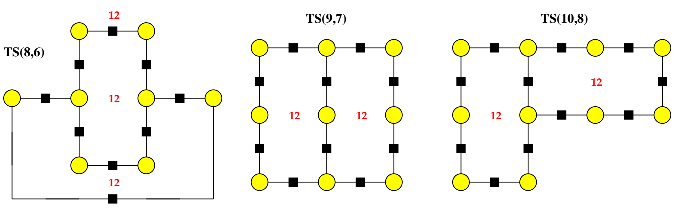

Fig. 1: Smallest TSs that would stop the decoder with undetected errors under IBL-PS(1)

We have specifically designed a QC-LDPC code with desirable properties for the IBL-PS(1) criterion using the design algorithm of [10]. The designed QC-LDPC code has a $(M_b, N_b) = (3, 6)$ array-type base matrix B, with $b_{i,j} = 1$ $\forall i, j$. The circulant size is $L = 128$ which results in an expanded code length $N = 768$ bits. The coding rate is $R = 1/2$. Table I shows the cycle distribution of H and the three associated submatrices. We can see that although the global girth of the LDPC graph is $g = 10$, it is possible to ensure that the combination of two layers form subgraphs with girth $g = 12$.

IV. Performance Comparison with Other Works

There has been plenty of low-complexity imprecise stopping criteria proposed in the literature. The ones that have the smallest implementation complexity are the on-the-fly syndrome check (OTF) proposed in [6], and the *sign stability* (SS) proposed in [2]. Other techniques, based on monitoring the amplitude variations of the AP-LLR messages [3], [8], [9], require an extra load of hardware resource and are not considered in this letter.

The SS criterion monitors the signs of the AP-LLRs between consecutive layers. At each layer H_i, the signs of the AP-LLRs before and after layer processing are compared. If all signs match, then a counter is incremented by 1, otherwise the counter is reset. Decoding stops when the counter reaches a value θ (in [2], $\theta = M_b$). We will denote this criterion SS(θ). Although this technique provides a stopping criterion which is almost safe, it typically increases the average number of iterations needed to stop the decoder.

Similarly, for the OTF criterion, a counter is incremented by 1 if a partial syndrome $\boldsymbol{s}_i$ is satisfied, and reset to zero if not satisfied. The decoding stops when the counter reaches θ. We will denote this criterion OTF(θ). In [6], the authors fixed $\theta = M_b$ and the counter was reset at the beginning of each iteration, while in [7], the authors considered a safer condition with $\theta > M_b$. The weakness of the OTF(θ) syndrome comes from the fact that the hard decision vector $\hat{\boldsymbol{x}}$ may change from one layer to another. Therefore, oscillations in the decision vector are not detected by OTF, which renders it unsafe.

As a first demonstration of the performance of our stopping criterion, we have made a comparison in terms of error correction performance between the IBL-PS(1), the IBL-PS(2), the SS(3) and the OTF(θ) with $\theta = \{3, 6, 9\}$ and the full syndrome stopping criterion, for the QC-LDPC code reported in Table I. The decoding results are reported in Figure 2, for the additive white Gaussian noise (AWGN) channel with signal to noise ratio E_b/N_0. The offset Min-Sum decoder has been used [1], with layered scheduling and a maximum of 50 decoding iterations.

We note that the IBL-PS(1) stopping criterion does not introduce any performance loss compared to the full syndrome until the frame error rate (FER) reaches an error floor at FER $\approx 10^{-4}$. All frame errors in the error floor of the curves are undetected errors. It can also be seen that the IBL-PS(2) is as safe as the SS(3), with no performance degradation compared with the full syndrome check. As expected, the OTF(θ) is not safe, as it introduces a significant degradation of the error correction performance. The OTF(9) has similar performance than the other stopping criteria down to FER=10^{-6}, which could be sufficient for some applications.

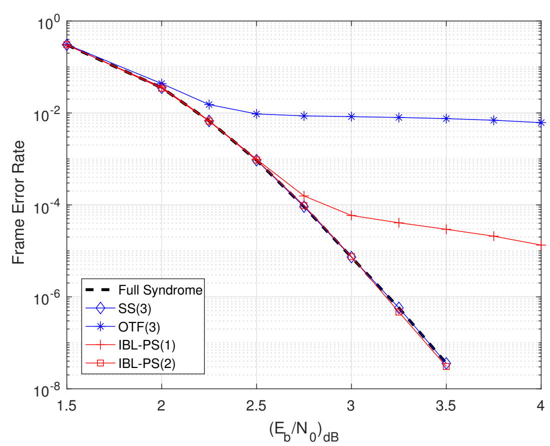

Fig. 2: Different stopping criteria on a $(M_b, N_b) = (3, 6)$, $N = 768$ QC-LDPC code

In Figure 3, we compare the average number of iteration for code in Table I, using different stopping criteria that have the same error correction performance, *i.e.* Syndrome based,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2017.2718523, IEEE Communications Letters

E_b/N_0	1.50	1.75	2.00	2.25	2.50	2.75	3.00
$\overline{It}$	11.18	8.14	6.79	5.94	5.32	4.84	4.47

TABLE II: Convergence Rate for the Wimax Code, corresponding to Figure 4.

SS(3), OTF(9) and IBL-PS(2). Note that with the IBL-PS(θ), the SS(θ) and the OTF(θ), the decoder can stop at the end of a layer processing and not necessarily at the end of an iteration. We note that the full syndrome check is not actually an *on-the-fly* stopping criterion and is only included here for comparison purposes. As a matter of facts, in practical implementations the full syndrome is computed in parallel with the processing of the next decoding iteration, thus resulting in a penalty of $+1$ iteration for stopping the decoder. This has been taken into account in Figure 3. We can see that the IBL-PS(2) stopping criterion allows reducing the average number of iterations by up to 30% compared to the SS(3) criterion (which is also safe), and up to 108% compared to the OTF(9) criterion (which is not safe below FER=10^{-6}).

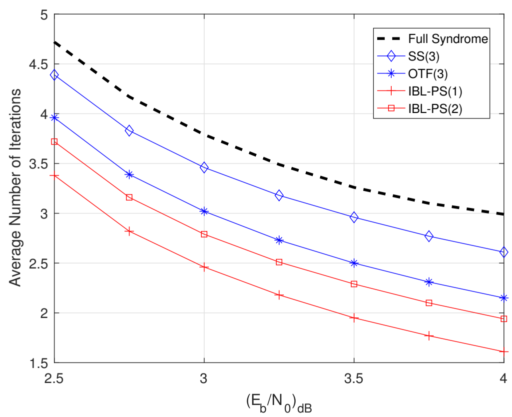

Fig. 3: Convergence rate for different stopping criteria and QC-LDPC code $(M_b, N_b) = (3, 6)$, $N = 768$

In a second example, we compare the stopping criteria on the protograph QC-LDPC codes from the WIMAX standard [13], which is an irregular code with code length $N = 2304$ and $M_b = 12$ layers. Note that for this code, the layers contain all-zero blocks, and we therefore expect the imprecise stopping criteria to be less safe than in the previous example. For this code, we have computed the average number of iterations for all stopping criteria with different values of θ. For each criterion, we choose the value of θ such that the average number of iterations matches the one of the full syndrome, indicated in Table II. With this setting, we can compare the safeness of each stopping criterion at equal latency. Results are reported on Figure 4. As we can see, the only criterion which approaches the safeness of the full syndrome is the IBL-PS(5).

V. Conclusion

In this letter we addressed the issue of early stopping criterion for layered QC-LDPC decoders, aiming at low hardware cost, minimum latency and improved safeness. We have introduced a new on-the-fly measure in the decoder, called in-between layers partial syndrome, and defined a family of stopping criteria using this new measure, with different tradeoffs between latency and performance. We show that our imprecise stopping criterion is sufficiently safe to be considered in practical applications, while surpassing existing solutions from the state-of-the-art.

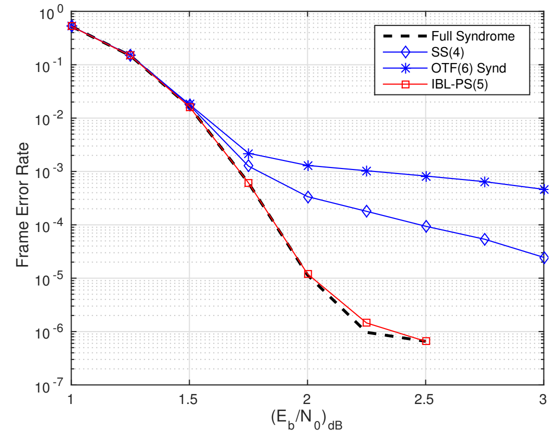

Fig. 4: Safeness comparison at equal average latency for the WIMAX code. $R = 1/2$ and $N = 2304$.

References

[1] D. Declercq, M Fossorier, E Biglieri (Editors), "Channel Coding: Theory, Algorithms, and Applications", Academic Press Library in Mobile and Wireless Communications, Elsevier, ISBN: 978-0-12-396499-1, 2014.

[2] J. Li, G. He, H. Hou, Z. Zhang and J. Ma, "Memory Efficient Layered Decoder Design with Early Termination for LDPC Codes", Proc. of the IEEE Int. Symp. On Circuits and System (ISCAS), Rio de Janeiro, Brazil, May 2011.

[3] M. Ferrari, S. Belini and A. Tomasoni, "Safe Early Stopping for Layered LDPC Decoding", IEEE Communications Letters, vol. 19, no. 3, pp. 315–318, March 2015.

[4] M.P.C. Fossorier, "Quasicyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices", IEEE Trans. on Information Theory, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[5] D.E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes", Proc. of the IEEE Workshop on Signal Processing Systems (SIPS), Austin, Texas, USA, Oct. 2004.

[6] E. Amador, R. Knopp, R. Pacalet and V. Rezard, "On-the-fly Syndrome Check for LDPC Decoders", Proc. of the 6th Int. Conf. on Wireless and Mobile Communications (ICWMC), Valencia, Spain, Sept. 2010.

[7] A. Hera, O. Boncalo, C.-E. Gavriliu, A. Amaricai, V. Savin, D. Declercq and F. Ghaffari, "Analysis and implementation of on the fly stopping criteria for layered QC LDPC decoders", Proc. of the IEEE Int. Conf. on Mixed Designs of Integrated Circuits and Systems (MIXDES), Torun, Poland, June 2015.

[8] C.H. Lin, T.H. Huang, C.C. Chen and S.Y. Lin, "Efficient layer stopping technique for layered LDPC decoding", Electronic Letters, vol. 49, no. 16, pp. 994–996, Aug. 2013.

[9] Y. Sun and J.R. Cavallaro, "A low-power 1-Gbps reconfigurable LDPC decoder design for multiple 4G wireless standards", Proc. of the 28th IEEE Int. Conf. on Systems on Chips (SOC), Newport Beach, USA, Sept. 2008.

[10] A. Venkiah, D. Declercq and C. Poulliat, "Design of Cages with a Randomized Progressive Edge Growth Algorithm", IEEE Communications Letters, vol. 12, no. 4, pp. 301–303, April 2008.

[11] G. Liva, S. Song, L. Lan, Y. Zhang, S. Lin and W. Ryan, "Design of LDPC Codes: A Survey and New Results," Journal of Communications Software and Systems, vol. 2, no. 3, pp. 191–211, September 2006.

[12] C. Poulliat, M. Fossorier and D. Declercq, "Design of regular (2,dc)-LDPC codes over GF(q) using their binary images", IEEE Trans. Commun., vol. 56, no. 10, pp. 1626–1635, October 2008.

[13] IEEE-802.16e, *in Physical and Medium Access Control Layers for Combined Fixe and Mobile Operation in Licensed Bands, amendment to Air Interface for Fixed Broadband Wireless Access Systems*, 2005.