Link between Sum-Product and Gradient Projection Decoding of LDPC codes: an Intermediate Algorithm

Maxime Beaudonnet, Christos Kasparis and Barry G. Evans
Centre For Communication Systems Research (CCSR), University of Surrey, Guildford GU2 7XH, UK
Email: m.beaudonnet@surrey.ac.uk, c.kasparis@surrey.ac.uk, b.evans@surrey.ac.uk

***Abstract*—This paper investigates the connection between the classical Sum-Product (SP) decoder for Low Density Parity Check (LDPC) codes and the recently proposed Gradient Projection (GP) decoding scheme presented in [1]. A graphical model for GP is exhibited based on which we derive an intermediate algorithm which establishes a bridge between graphical based algorithms (SP and variants) and an optimization based algorithm (GP). A more practical decoding algorithm with improved performance and reduced complexity is also proposed. A complexity analysis is provided and performance are studied through Monte-Carlo simulations.**

I. INTRODUCTION

LDPC codes are now established as one of the most powerful classes of error correcting codes. Noticeably, the DVB-S2 standard FEC scheme relies on a set of capacity approaching LDPC codes. They are also present in the Wimax (802.16e) and Wifi (802.11n) standards.

Near optimal decoding is usually achieved with the Sum-Product (SP) algorithm which is also recommended in the DVB-S2 standard. Essentially, the SP algorithm computes A Posteriori probabilities of code bits if the Tanner graph associated to the LDPC code is cycle free. Numerous simulations have shown that the performance penalty induced by cycles is reasonnable. The most common low complexity variant of SP is the Min-Sum (MS) algorithm which typically performs 0.5 dB away [6].

Other decoding schemes have been proposed (Linear Programming [9], Non-Linear Programming [10], Bit Flipping [7]) which aim at finding the Maximum Likelihood codeword without explicitely computing probabilities.

The GP decoder has recently been introduced in [1] as an alternative decoding approach based on formulation and optimization of an objective function which entails the code constraints. Convex constrained optimization is achieved with the GP iteration [8] and aim at finding a global extremum at a vertice of the hypercube $[0,1]^n$ that, by construction, corresponds to a valid codeword.

We derive in this contribution a graphical model for GP which can thus be viewed as a type of message passing algorithm similarly to SP. Based on its graphical interpretation, two variants of GP are derived. The first is not of practical interest but highlights the connection between GP and SP. The second algorithm is practical for it is simpler and better performing than the original GP.

Section II provides the background on LDPC codes and the notations used in the paper. Section III describes the SP and the GP decoders and IV the intermediate algorithms and their connection to them. Section V gives several simulation results to assess the performances of the algorithms discussed here. Section VI concludes this paper.

II. BACKGROUND AND NOTATIONS

LDPC codes are a special class of linear block codes. A linear block code can be described by its parity-check matrix $H = (h_{ji})$, of dimension (m, n). LDPC codes are characterized by very sparse parity-check matrices. The sparsity constraint keeps the decoder complexity manageable even for very long codes, which need to be employed if Shannon bound is to be approached. Based on H, we define several sets that will be used throughout the paper: $\mathcal{I} = \{1 \cdots n\}$, $\mathcal{J} = \{1 \cdots m\}$, for every $j \in \mathcal{J}$, $\mathcal{I}_j = \{i \in \mathcal{I} | h_{ji} = 1\}$, for every $i \in \mathcal{I}$, $\mathcal{J}_i = \{j \in \mathcal{J} | h_{ji} = 1\}$ and $\mathcal{E} = \{(i,j) \in \mathcal{I} \times \mathcal{J} | h_{ji} = 1\}$. A Tanner graph provides an alternative way to represent H. It is composed of a set of n variable nodes (V-nodes) indexed by $\mathcal{I}$ and a set of m check nodes (C-nodes) indexed by $\mathcal{J}$. The edges of the Tanner graph are indexed by $\mathcal{E}$. An edge (i,j) connects V-node i and C-node j if and only if code bit i is involved in parity-check equation j.

In the paper, it is assumed a BPSK modulation of code bits $(c_i)_{i=1..n}$ into symbols $(s_i)_{i=1..n}$ $(s_i = 1 - 2.c_i)$ and an AWGN channel. The channel observation is noted $(y_i)_{i=1..n}$.

The SP algorithm consists of iteratively passing messages along the edges connecting V-nodes and C-nodes. The messages convey information about the likelihood of the code bits being 1 or 0. The message concerning a code bit b can be expressed in three different metrics (see [4]): probabilities ($Pr(b = 0)$ or $Pr(b = 1)$), Log-Likelihood Ratios ($LLR(b) = \ln(\frac{Pr(b=0)}{Pr(b=1)})$) or soft-bits ($B = \tanh(\frac{LLR(b)}{2})$). For the description of the graph-based algorithms, we use the following notations for the messages passing on the graph model: v_{ij}^k (LLR) and V_{ij}^k (soft-bit) denote the message passing along edge (i,j), from V-node i to C-node j at iteration k. Similarly, the message from C-node j to V-node i is noted c_{ji}^k (LLR) or C_{ji}^k (soft-bit). The estimate at iteration k of the A Posteriori probability of code bit i is noted v_i^k (LLR) or V_i^k (soft-bit) and the A Priori probability derived from the channel observation is noted f_i (LLR) or F_i (soft-bit). With

978-1-4244-1645-5/08/$25.00 ©2008 IEEE

the channel assumptions made above, the latter quantities are expressed by: $f_i = \frac{2.y_i}{\sigma^2}$ and $F_i = \tanh\left(\frac{y_i}{\sigma^2}\right)$ where σ^2 is the noise variance.

III. Sum-Product algorithm and Gradient Projection decoding

A. Sum-Product and Min-Sum algorithms

In order to illustrate the connection between GP and SP in section IV, we will use in this section both LLR and soft bit metrics to describe SP. The simplification of SP into MS is also discussed. Both SP and MS are iterative algorithms. An iteration consists of three parts: the C-nodes and V-nodes processing and the decision variable update. At the end of each iteration, a candidate codeword is derived from the decision variable. If the syndrom of that candidate is null, the algorithm is stopped as the codeword is valid. In order to handle the case of decoding failure, the algorithm is also stopped after a user-defined number of iteration. This stopping criterion is assumed for all the algorithms described in the paper. The SP algorithm is summarized below.

```
for all $i \in \mathcal{I}$ do
  for all $j \in \mathcal{J}_i$ do
      $v_{ij}^0 = y_i$
      $V_{ij}^0 = Y_i = \tanh\left(\frac{y_i}{2}\right)$
while stopping criterion not satisfied do
  Step 1: C-nodes processing
  for all $j \in \mathcal{J}$ do
    for all $i \in \mathcal{I}_j$ do
      $c_{ji}^k = 2 \cdot \text{atanh}\left(\prod_{\substack{i' \in \mathcal{I}_j \\ i' \neq i}} \tanh\left(\frac{v_{i'j}^{k-1}}{2}\right)\right)$
      $C_{ji}^k = \prod_{\substack{i' \in \mathcal{I}_j \\ i' \neq i}} V_{i'j}^{k-1}$
  Step 2: V-nodes processing
  for all $i \in \mathcal{I}$ do
    for all $j \in \mathcal{J}_i$ do
      $v_{ij}^k = y_i + \sum_{\substack{j' \in \mathcal{J}_i \\ j' \neq j}} c_{j'i}^k$
      $V_{ij}^k = \tanh\left(\text{atanh}(Y_i) + \sum_{\substack{j' \in \mathcal{J}_i \\ j' \neq j}} \text{atanh}\left(C_{j'i}^k\right)\right)$
  Step 3: Decision variables update
  for all $i \in \mathcal{I}$ do
      $v_i^k = y_i + \sum_{j \in \mathcal{J}_i} c_{ji}^k$
      $V_i^k = \tanh\left(\text{atanh}(Y_i) + \sum_{j \in \mathcal{J}_i} \text{atanh}\left(C_{ji}^k\right)\right)$
  Step 4: stopping criterion
```

In the LLR metrics variant, which is usually implemented, complexity is driven by the C-nodes processing. The MS simplification classically applied to SP consists in replacing this processing by:

$$c_{ji}^k = \prod_{\substack{i' \in \mathcal{I}_j \\ i' \neq i}} a_{i'j}^{k-1} \cdot \min_{\substack{i' \in \mathcal{I}_j \\ i' \neq i}} \left(b_{i'j}^{k-1}\right)$$

where messages $v_{i'j}^k$ are factorized into sign and magnitude: $v_{i'j}^k = a_{i'j}^k \cdot b_{i'j}^k$ with $a_{i'j} = \text{sign}(v_{i'j})$ and $b_{i'j} = |v_{i'j}|$. The MS simplification is widely adopted and typically performs 0.5 dB away from the SP algorithm.

B. Gradient Projection Decoding

GP is a recently proposed approach for decoding LDPC codes in which the decoding problem is formulated as a convex constrained optimization that is solved by the classical gradient projection algorithm (see [1] and [8]). The cost function introduced below differs from the original one ([1]). It is simpler for it has no longer a nested form and reduces the computationnal load at no cost in performance. This function depends on the n variables representing soft estimates of the code bits and is constructed such that it entails the m parity check equations. For $i \in I$, we denote in the following x_i^k the soft value representing an estimate of symbol s_i, at iteration k. The cost function associated to some parity-check equation j is:

$$f_j\left((x_i)_{i \in \mathcal{I}_j}\right) = \prod_{i \in \mathcal{I}_j} x_i$$

Since a parity-check equation is satisfied if and only if an even number of bits involved are $'1'$s, we can affirm that when all x_i are in $\{-1, 1\}$, the parity check equation j is satisfied if and only if $f_j = 1$ (recall that code bit c_i is $'1'$ if and only if s_i, or equivalently x_i is $'-1'$). An overall cost function is thus constructed as follows:

$$F(\mathbf{x}) = \sum_{j=1}^{m} f_j \text{ with } \mathbf{x} = [x_1 \cdots x_n] \in [-1, 1]^n$$

The classical Gradient Projection algorithm, applied over the set $[-1, 1]^n$, enables to perform the maximization. The variables x_i are initialized with the soft output of the AWGN channel y_i and the GP algorithm modifies the x_i values until the closest local maximum of the cost function is found. Let us denote u the projection operator over $[-1, +1]$: $\text{u}(x) = \text{sign}(x)$ if $|x| \geq 1$ and $\text{u}(x) = x$ if $|x| < 1$. Gradient Projection decoding is described hereafter:

```
for all $i \in \mathcal{I}$ do
    $x_i^0 = \text{u}(y_i)$
while stopping criterion not satisfied do
  Step 1: Gradient computation and soft information update
  for all $n \in \mathcal{I}$ do
```

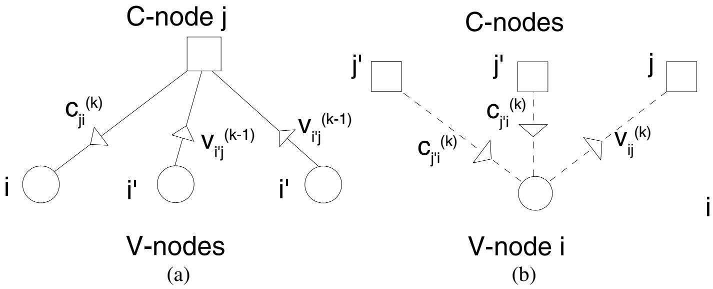


Fig. 1. Graph description of the SP algorithm: (a) C-node processing (b) V-node processing

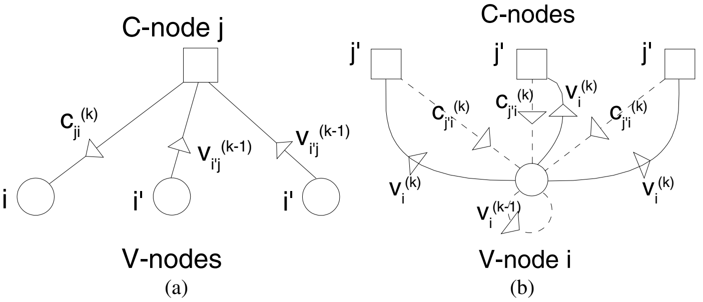


Fig. 2. Graph description of the proposed algorithm: (a) C-node processing (b) V-node processing

$$\frac{\partial F}{\partial x_i} = \sum_{j=1}^{m} \frac{\partial f_j}{\partial x_i} = \sum_{j\in\mathcal{J}_i}\left(\prod_{\substack{i'\in\mathcal{I}_j\\ i'\neq i}} x_{i'}^{k-1}\right)$$

$$x_i^k = \mathrm{u}\left(x_i^{k-1}+\alpha\cdot\frac{\partial F}{\partial x_i}\right)$$

Step 2: stopping criterion

The soft values x_i are used as decision variables. The optimization theory assures that this algorithm will converge toward a local maximum, which however does not always correspond to a valid codeword (only global maxima are in one to one correspondance with valid codewords). The choice of a value for the step parameter α is a trade-off between accuracy and convergence speed. If α is chosen sufficiently small, the algorithm will always converge (see [8]).

Simulations show that the GP decoding method is suboptimal. It is however a promising new decoding technique for it is significantly less complex.

IV. Connection between SP and GP: intermediate algorithms

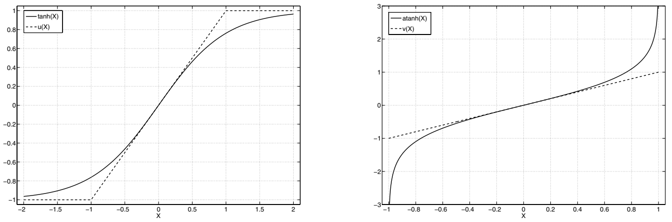
This section sheds light on the connection between SP and GP decoders. In order to exhibit tight algorithmic links, we introduce an intermediate algorithm (called modified SP) in section IV-A. It is further shown that the piece-wise linearization of this algorithm reduces to the original GP algorithm, highlighting the connection between GP and SP. Though this algorithm does improve the performance of GP, it is not claimed to be a practical decoding alternative as it is more complex than GP. However, it is shown in section IV-B that the classical MS simplification is applicable to it and that the resulting algorithm becomes less complex and better performing than GP. The latter algorithm, called modified Min-Sum, is claimed to be a practical alternative for decoding LDPC codes.

A. Modified Sum-Product

The modification of SP algorithm proposed in this section is to give up the extrinsic information principle. Instead of updating a set of messages v_{ij} and a decision variable v_i^k, we only store and update an approximate A Posteriori Probability, and this approximation is used as the message itself. Graphically, the original SP algorithm and the proposed modification are illustrated on Fig.1 and Fig.2. Only the V-node processing and the decision variable update are modified with regard to the standard SP algorithm and this modification is inspired by the GP algorithm. The proposed algorithm is described below.

```
**for all** $i \in \mathcal{I}$ **do**
  $v_i^0 = y_i$
  $V_i^0 = Y_i = \tanh\left(\frac{y_i}{2}\right)$
**while** stopping criterion not satisfied **do**
  *Step 1: C-nodes processing*
  **for all** $j \in \mathcal{J}$ **do**
    **for all** $i \in \mathcal{J}_i$ **do**
      $c_{ji}^k = 2\cdot\operatorname{atanh}\left(\prod_{\substack{i'\in\mathcal{I}_j\\ i'\neq i}} \tanh\left(\frac{v_{i'}^{k-1}}{2}\right)\right)$
      $C_{ji}^k = \prod_{\substack{i'\in\mathcal{I}_j\\ i'\neq i}} V_{i'}^{k-1}$
  *Step 2: V-nodes processing and decision variables update*
  **for all** $i \in \mathcal{I}$ **do**
      $v_i^k = v_i^{k-1} + \alpha\cdot\sum_{j\in\mathcal{J}_i} c_{ji}^k$
      $V_i^k = \tanh\left(\operatorname{atanh}(V_i^{k-1}) + \alpha\sum_{j\in\mathcal{J}_i}\operatorname{atanh}\left(C_{ji}^k\right)\right)$
  *Step 3: stopping criterion*
```

The above algorithm can be view as an intermediate one between SP and GP. The SP algorithm and its links to the factor graph theory are described in [3]. The proposed algorithm respects the behavioural model described in [3] and thus has a graphical interpretation. However, it does not comply with the probabilistic model as it does not produce an exact APP distribution contrarily to SP on a cycle free graph. The proposed modified SP algorithm repeatedly invokes the $\tanh$ and atanh functions. If we replace these two functions



(a) (b)

Fig. 3. Linearization: (a) of the $\tanh$ function (b) of the atanh function

by the piece-wise linear function u and the identity function over $[-1,+1]$ (see Fig.4) in the modified SP expressed in soft-bit metrics, we obtain the following algorithm:

```
for all $i \in \mathcal{I}$ do
    $V_i^0 = Y_i = \mathrm{u}\left(\frac{y_i}{2}\right)$
while stopping criterion not satisfied do
    Step 1:
    for all $i \in \mathcal{I}$ do
        $V_i^k = \mathrm{u}\left(V_i^{k-1} + \alpha \cdot \sum_{j \in \mathcal{J}_i} \prod_{\substack{i' \in \mathcal{I}_j \\ i' \neq i}} V_{i'}^{k-1}\right)$
    Step 2: stopping criterion
```

It is interesting to note that the above algorithm is the original GP algorithm except for the initialization step. Simulations show that GP performs better if initialized with the channel observation y_i. This piece-wise linearization establishes a bridge between a degraded SP algorithm and GP, giving further insight into the recent GP decoding of LDPC codes. Though the proposed intermediate algorithm relates SP and GP decoding, it comes with an increased complexity with regard to GP. However, it is shown in section IV-B that a practical algorithm can be obtained by applying the classical MS simplification to the C-nodes processing.

B. Modified Min-Sum algorithm

The modified SP algorithm presented in the previous section involves the same C-node processing as the original SP. It is then natural to apply the classical MS simplification. Simulations show that the resulting modified MS has reduced complexity and improved performance with regard to GP. In order to support the claim that the proposed modified MS algorithm is a practical decoding alternative, we conduct in the following a complexity analysis in order to compare the original SP and GP algorithms to the proposed modified MS.

The complexities of these algorithms are summarized in TABLEI. For the complexity comparison, we have considered a regular LDPC code of C-node degree $\rho = |\mathcal{I}_j|$ and of V-node degree $\gamma = |\mathcal{J}_i|$. The 'Min.' operation corresponds to finding the two lowest values among a set of ρ values which correponds to the C-nodes processing in MS and modified MS. The complexity of a decoder is also determined by the amount of storage space required. We only consider in the following analysis the storage of real numbers. Hence we count only one storage unit for the case where several messages have the same magnitude but different signs. The MS and GP algorithms typically require the storage of $\mathrm{O}(n.(\gamma+4))$ and $\mathrm{O}(2.n)$ real numbers respectively . The modified MS algorithm requires $\mathrm{O}(3.n)$ real numbers to be stored.

	Standard Min-Sum	Modified Min-Sum	Gradient Projection
Multiplication	X	$\mathrm{O}(n)$	$\mathrm{O}(n.\gamma.\rho)$
Addition	$\mathrm{O}(n.\gamma.\gamma)$	$\mathrm{O}(n.\gamma)$	$\mathrm{O}(n.\gamma)$
Min. operation	$\mathrm{O}(n.\gamma)$	$\mathrm{O}(n.\gamma)$	X
Projection	X	X	$\mathrm{O}(n)$
Sign inversion	$\mathrm{O}(n.\gamma.\rho)$	$\mathrm{O}(n.\gamma.\rho)$	X
Storage required	$\mathrm{O}(n.(\gamma+4))$	$\mathrm{O}(3.n)$	$\mathrm{O}(2.n)$

TABLE I
COMPLEXITY COMPARISON

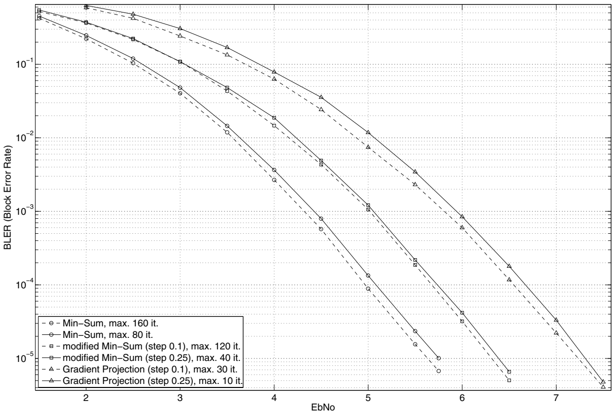


Fig. 4. BLER performance of a (96,48) LDPC code

For the sake of fair comparison, one should also consider the typical number of iterations required to achieve best performance. To that respect, a desirable feature of GP and modified MS is the existence of the step parameter which can be tuned to increase the convergence speed. Quantified results are given in the next section. The general rule is that GP is the fastest decoder and MS the slowest.

Overall, the proposed modified Min-Sum decoder is the simplest algorithm of the three compared in the paper even though the GP decoder ensures the fastest decoding which is of interest for low latency applications. The best performing algorithm remains the classical MS.

V. Simulation results

One irregular and two regular LDPC codes have been used to conduct block error rate (BLER) Monte-Carlo simulations. For GP and modified MS, we show that convergence can be sped up by increasing the step parameter. The performance gaps given below are at BLER 10^{-5} and compare the best performing variants of each algorithm (dashed lines on Fig.4, Fig.5 and Fig.6).

Results obtained with the shorter regular code ($n = 96$, $k = 48$, $\rho = 6$ and $\gamma = 3$) are shown on Fig.4. The modified MS algorithm performs 0.7 dB worse than the original MS

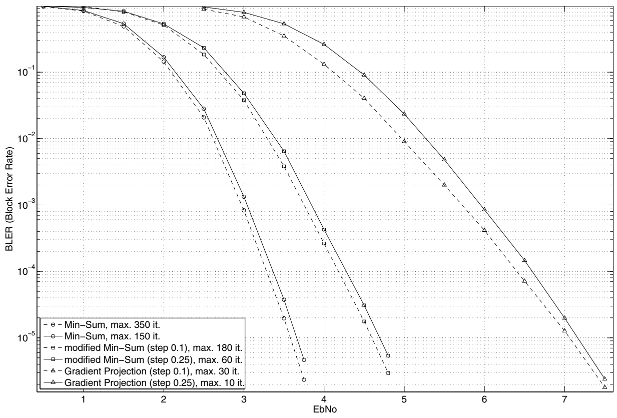


Fig. 5. BLER performance of a (504,252) LDPC code

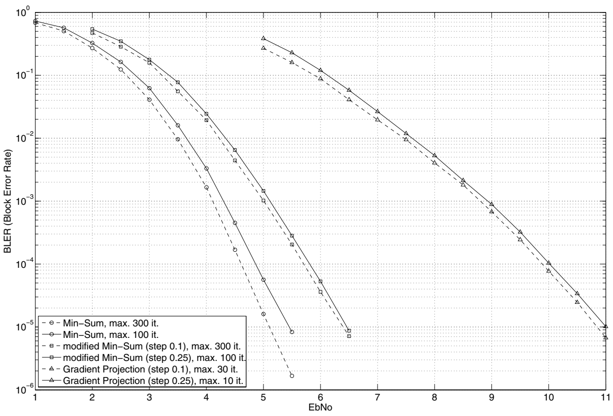


Fig. 6. BLER performance of a (200,52) LDPC code

and 0.5 dB better than GP. The results in solid lines show that the algorithms can be stopped significantly earlier with a low performance penalty. Noticeably, the modified MS and GP can be stopped after few iterations (40 and 10 respectively) and are therefore good candidates for low latency applications.

Performance results for the longer regular code ($n = 504$, $k = 252$, $\rho = 6$ and $\gamma = 3$) are given on Fig.5. On this code, the modified MS algorithm and GP trail the MS algorithm by 1 dB and 3.5 dB respectively which means that the proposed modified MS brings a 2.5 dB improvement together with a complexity reduction when compared to GP. However, GP remains the fastest converging algorithm with a sufficient maximum number of 10 iterations to be compared to the required 60 and 150 for modified MS and MS respectively.

An irregular ($n = 200$ and $k = 52$) LDPC code has also been simulated. Results are shown on Fig.6. Modified MS outperforms GP by 4.4 dB but performs 1.3 dB worse than MS. Note that on this code, only GP provides a significant advantage in decoding speed (with a maximum of 10 iterations).

Simulations show that the introduced algorithm significantly reduces the performance gap between GP and MS and is often faster (and at least not slower) than the classical MS algorithm. These performance improvements come and support the graphical interpretation of GP introduced in this paper.

VI. Conclusion

The connection between the recent GP decoding of LDPC codes and the classical SP (and its variants) has been investigated. The main contribution is the construction of an underlying graphical model for GP which establishes a bridge between SP related algorithms and an optimization based algorithm. The connection between two seemingly different classes of algorithms has been highlighted by the introduction of an intermediate algorithm which is essentially a degraded SP algorithm and which reduces to the original GP after piecewise linearization. The existence of a graphical interpretation to GP gives a new lead for further improvements of this low complexity decoding of LDPC codes. Furthermore, based on the introduced graph model, a practical algorithm has been derived which is of interest for low complexity, low latency applications (typically handheld terminals). It exhibits significant performance improvement together with a reduced complexity with regard to GP, at the cost of a reasonnable increase in decoding delay. It suffers from reasonnable performance degradation when compared to the classical MS. The adoption of LDPC codes in the DVB-S2 standard make them good candidates for future satellite mobile applications for which the existence of fast and simple alternatives to the existing decoders is a critical requirement. The work conducted in this paper is one step toward the design of such alternatives.

Acknowledgment

This paper has been supported by the European IST-FP6 project: "Satnex II-Satellite Communications Network of Excellence II".

References

[1] C. Kasparis and B. G. Evans, "Gradient Projection Decoding of LDPC codes", *IEEE Communications letters*, Vol. 11, No. 3, March 2007.

[2] S. Lin and D. J. Costello Jr, "Error Control Coding", Second Edition, *Pearson Prentice Hall*, 2004.

[3] F. R. Kschischang, B. J. Frey, and H. A Loeliger, "Factor Graphs and the Sum-Product Algorithm", *IEEE Transactions on Information Theory*, Vol. 47, No. 2, February 2001.

[4] S. L. Howard, V. C. Gaudet, C. Schlegel, "Soft-Bit Decoding of Regular Low-Density-Parity-Check Codes", *IEEE Transactions on circuits and systems-II: Express briefs*, Vol. 52, No. 10, October 2005.

[5] R. G. Gallager, "Low-Density parity-check codes", Cambridge, MA: MIT Press, 1963

[6] J. Chen, M. P. C. Fossorier, "Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes", *IEEE Transactions on Communications*, Vol. 50, No. 3, March 2002.

[7] Z. Liu, D. A. Pados, "A Decoding Algorithm for Finite-Geometry LDPC Codes", *IEEE Transactions on Communications*, Vol. 53, No. 3, March 2005.

[8] E. S. Levitin and B. T. Polyak, "Constrained minimization methods", *USSR computational Mathematics and Mathematical Physics*, vol. 6, pp. 1-50, 1966.

[9] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using Linear Programming to Decode Binary Linear Codes", *IEEE Transactions on Information Theory*, Vol. 51, No. 3, March 2005.

[10] K. Yang, J. Feldman, and X. Wang, "Nonlinear Programming Approaches to Decoding Low-Density Parity-Check Codes", *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 8, August 2006.