Fast ML Decoding of SPC Product Code by Linear Programming Decoding

Kai Yang and Xiaodong Wang
Department of E.E.
Columbia University
New York, NY,10027
{kyee,wangx}@ee.columbia.edu

Jon Feldman
Google New York
NY, 10010
jonfeld@google.com

***Abstract*— We consider the maximum-likelihood decoding of single parity-check (SPC) product code. We first prove that, for the family of SPC product code, the fractional distance and the pseudo-distance are both equal to the minimum Hamming distance. We then develop an efficient algorithm for decoding SPC product codes with low complexity and near maximum likelihood decoding performance at practical SNRs.**

I. Introduction

The linear Programming (LP) decoding [1], [2], [3] opens up a new avenue for the decoding of general linear block codes. The solid theoretic foundation of LP makes it simpler to analyse than the conventional message-passing decoders. In addition, the concept of pseudo-codeword arising from the LP decoding can be used to characterize the performance of iterative decoding algorithms such as the sum-product algorithm [4], [5], [6], [7].

An interesting yet open problem in coding theory is whether there exist any practical "good" codes which are maximum-likelihood decodable under low-complexity decoding algorithms such as the sum-product algorithm or LP decoding method. The term "good" here has three different meanings. First, the code must be able to achieve near capacity performance. Second, it should has low encoding complexity. Thirdly, it must be able to provide a wide range of code rates and block sizes.

It is known that pseudo-codewords account for decoding failures of iterative algorithms based on message-passing or linear programming. The LP decoder is guaranteed to output a ML codeword if the LP polytope (or fundamental polytope) is exactly the same as the code polytope. However, it seems this condition is too conservative, and so far no "good" codes satisfying this condition have been found. A better alternative is to consider codes with pseudo-distance equal to minimum Hamming distance. Indeed, it has been shown that if the pseudo-weight spectrum gap for a code is strictly positive, the performance of the LP decoder approaches that of the ML decoder when the SNR goes to infinity [8].

To goal of this paper is to present such a class of "good" codes and a corresponding fast ML decoding algorithm. We show that, for the family of SPC product codes, the fractional distance and pseudo-distance are both equal to the minimum distance. Inspired by this fact, we develop an efficient subgradient decoder that achieves near ML decoding performance for SPC product codes. To the best of our knowledge, this is the first group of capacity approaching codes [9] which have been shown to be practically maximum-likelihood (ML) decodable under efficient decoding methods inspired by LP decoding.

The rest of this paper is organized as follows. Section 2 reviews the linear programming decoding and the definition of pseudo-codewords, fractional distance and pseudo-distance. In Section 3, we show that the fractional distance and pseudo-distance of the SPC product code are both equal to the Hamming distance. We propose an efficient decoder based on dual decomposition method for the SPC product code in Section 4. The simulation results are shown in Section 5. Section 6 contains the conclusion.

II. LP Decoding

Let $\boldsymbol{x}$ denote the transmitted codeword over the noisy channel, and $\boldsymbol{y}$ denote the received codeword. Given $\boldsymbol{y}$, the ML decoding problem is to obtain the estimated codeword that maximizes the likelihood of observing $\boldsymbol{y}$ under the channel model. Define ν_i as the log-likelihood ratio given by

$$\nu_i = \log \frac{\Pr[y_i|x_i=0]}{\Pr[y_i|x_i=1]}, \tag{1}$$

The ML decoding corresponds to finding the codeword $\boldsymbol{f} = [f_1, f_2, ..., f_n]^T$ that minimizes $\sum_{i=1}^{n} \nu_i f_i$ [3]. We may interpret ν_i as the "cost" of decoding f_i=1.

Denote the parity-check matrix of the binary linear code as $\boldsymbol{H} = [H_{ij}]_{m\times n}$, the integer programming formulation of the

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

ML decoding problem is then given by

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} \nu_j f_j \\ \text{s.t.} \quad & \left[\sum_{j=1}^{n} H_{ij} f_j\right]_2 = 0, \ \forall i, \quad \text{and} \quad f_j \in \{0,1\}, \quad \forall j \end{aligned} \quad (2)$$

where $[\cdot]_2$ denotes the modular-2 operation.

Assume there are γ_i ones in the i^{th} row of the parity-check matrix $\boldsymbol{H}$ with the index $[\beta_1, ..., \beta_{\gamma_i}]$. Thus the i^{th} parity-check constraint can be written as

$$\left[\sum_{k=1}^{\gamma_i} f_{\beta_k}\right]_2 = 0, \ f_j \in \{0,1\}, \ 1 \le j \le n. \quad (3)$$

Let $\Omega_i = \{\bar{\mathbf{f}} = [f_1, f_2, ..., f_n] : \bar{\mathbf{f}}$ satisfies constraints in $(3)\}$. The motivation of the LP decoding is to replace $\Omega_1, \Omega_2, ...\Omega_m$ in (2) by their corresponding convex hulls to obtain a linear programming problem, given by

$$\min \ \ \boldsymbol{v}^T \boldsymbol{f}, \ \ \text{s.t.} \ \boldsymbol{f} \in \text{conv}(\Omega_1) \cap \text{conv}(\Omega_2) ... \text{conv}(\Omega_m). \quad (4)$$

The set Ω_i is specified by the nonlinear parity-check constraint and integer constraints thus its convex hull cannot be obtained directly. One approach is to introduce auxiliary variables and replace the nonlinear parity-check constraint by a collection of linear constraints so that the corresponding convex hull can be obtained by simply relaxing the integer constraints. In [3], an LP formulation (F-LP) is proposed. But the number of auxiliary variables and constraints in this formulation increases exponentially with the degree of check nodes, which renders it impractical to linear codes with high-density parity-check matrices. In addition, for practical applications in which the code length is at least several hundred, the corresponding LP is too complex to solve even by using the efficient algorithms described in [10], [11]. A cascaded LP formulation (Y-LP) is proposed in [12]. In this formulation, the number of constraints and variables increase only linearly with the degree of check nodes.

III. Distance Property of the SPC product code

An SPC product code is a relatively large code built from smaller codeword blocks and its performance can be within 1 dB of the Shannon limit [14]. In addition, it has been shown that asymptotic probability of bit error can be driven to zero within 2 dB of the capacity of the additive white Gaussian noise channel [9] for an SPC product code. Without loss of generality, here we consider only the (n_s, d_s) SPC product code [14], where n_s is the length of its component code and d_s is its dimension. This code can be represented as a d_s-dimensional cube and the component code in each dimension is an SPC code with block length n_s. Thus the length of this code is $n_s^{d_s}$ and its rate is $\left(\frac{n_s-1}{n_s}\right)^{d_s}$. For example, the two-dimensional SPC product code consists of a data block, parity checks on rows, parity checks on columns and a check on checks. Please note that although the rate of this code approaches zero when the block length grows to infinity, it can achieve near capacity performance for practical block length and code rates [14]. In addition, asymptotic analysis has shown that there exist SPC product codes with non-zero rate which can approach the channel capacity when the block length goes to infinity [9].

Ideally, if we can get the convex hull of all codewords, i.e., $\text{conv}(\Omega_1 \cap \Omega_2 ... \Omega_m)$, then the ML codeword can be found by performing LP decoding over this convex hull. The following two results give examples of such an ideal parity-check matrix $\mathbf{H}$.

Proposition 1 *If the parity-check matrix satisfies* $\sum_{j=1}^{n} H_{i_1,j} H_{i_2,j} = 0$ *for some* i_1 *and* i_2*, then we have* $\text{conv}(\Omega_{i_1}) \cap (\Omega_{i_2}) = \text{conv}(\Omega_{i_1}) \cap \text{conv}(\Omega_{i_2})$.

Proposition 2 *If there exists at most one '1' in each column of the parity-check matrix* $\boldsymbol{H}$*, we have* $\text{conv}\left(\bigcap_{i=1}^{m} \Omega_i\right) = \bigcap_{i=1}^{m} \text{conv}\left(\Omega_i\right)$. *Thus the pseudo-codeword is the ML codeword*

The proofs are straightforward and thus omitted here. These two propositions can also be regarded as consequences of the Lemma 28 in [5], i.e., the linear programming relaxation is exact for trees.

However, in general, the polytope used in LP is not equivalent to the convex hull of the codewords, thus the ML decoding performance can not be obtained. For codes with small length, we can employ efficient integer programming techniques such as the branch-and-bound method to improve the decoding performance [10].

In [3], the fractional distance of an LDPC code is calculated by solving a group of LP problems. An objective function is minimized over the faces of the original polytope. The pseudo-distance [5] is a better metric to characterize the performance of an LP decoder. Denote by ℓ_f,ℓ_p and ℓ_m the fractional distance, pseudo-distance and minimum distance respectively. In general, we have $\ell_f \le \ell_p \le \ell_m$, and thus the performance of an LP decoder is inferior to that of the ML decoder. Here, we show theoretically that for SPC product codes, the fractional distance is equal to the minimum distance, i.e., $\ell_f = \ell_p = \ell_m$. In order to show this, the following two propositions are needed.

Proposition 3 *The fractional distance and pseudo-distance of any single parity-check code is equal to 2.*

Proof: The parity-check matrix of any single parity-check code contains exactly one "'1" in each column. Thus it follows from Proposition 2 that the LP polytope is equivalent to the code polytope. Therefore, its fractional distance and pseudo-distance are equal to 2, which is the minimum distance of any SPC code. ∎

Proposition 4 *Any feasible solution* $\boldsymbol{f}$ *to the LP decoding problem for any SPC code is either an all-zero codeword or it contains at least two non-zero bits, i.e.,* $f_i > 0,\ f_j > 0,\ 1 \leq i, j \leq n$.

Proof: Since the polytope of any SPC code is a convex hull, any point inside can be expressed as a convex combination of its extreme points. In addition, all points in the polytope are in the positive orthant. It follows from Proposition 3 that any non-zero extreme point contains at least two non-zero elements. Therefore, if **f** is not an all-zero codeword, it must contain at least two non-zero elements. ∎

Based on the above two propositions, we are ready to prove the following theorem.

Theorem 5 *The fractional distance and pseudo-distance of any* d_s*-dimensional SPC product code are equal to its minimum distance, i.e.,* $\ell_f = \ell_p = \ell_m = 2^{d_s}$.

Proof: It follows from Proposition 4 that the above theorem holds for $d_s = 1$. We will prove this theorem by induction. Suppose that it is true for $d_s = \eta$. An $(n_s, \eta + 1)$ SPC product code consists of n_s component codes where each component code is an (n_s, η) SPC product code. Since it is not an all-zero codeword, due to the assumption, at least one of these component codes is a non-zero pseudo-codeword and its minimum fractional distance is 2^{η}. On the other hand, it follows from Proposition 4 that there is at least another non-zero element outside this (n_s, η) component code, which leads to at least another non-zero (n_s, η) component pseudo-codeword. Hence, the fractional distance of an $(n_s, \eta+1)$ SPC product code is no less than twice the fractional distance of an (n_s, η) SPC product code, which equals to $2^{\eta+1}$. It is known that the minimum weight of an $(n_s, \eta+1)$ SPC product code is $2^{\eta+1}$ [14]. Thus, the fractional distance of any d_s-dimensional SPC product code is equal to 2^{d_s} and its pseudo-distance is equal to 2^{d_s}. ∎

The concept of pseudo-weight spectrum gap has been introduced in [8] to characterize the performance gap of the ML decoder and LP decoder. The pseudo-weight spectrum gap can be loosely defined as the difference between the smallest pseudo-weight of the non-codeword pseudo-codeword and the minimum distance. If the pseudo-weight spectrum gap is strictly positive, the performance of LP decoder will converge to that of the ML decoder when SNR grows to infinity. Since the fractional distance of the class of codes under consideration is equal to their minimum distance, it is easy to show that the minimum distance is smaller than the pseudo-weight of all other pseudo-codewords. We can therefore obtain the following theorem.

Theorem 6 *The pseudo-weight spectrum gap of the SPC product code is strictly positive.*

IV. A Fast LP Decoder for SPC Product Codes

Typically, general LP problems are solved by the interior-point method or the simplex method [15] and their variations. On the other hand, some codes exhibit specific structures that can be exploited to devise more efficient decoding algorithms. For example, low-complexity algorithms based on the F-LP formulation are proposed in [10], [11] for an LDPC code. In this section, we develop a fast decoding algorithm for SPC product code.

In [12], the Y-LP formulation is given by

$$\min \quad \boldsymbol{q}^T\boldsymbol{d}, \quad \text{s.t.} \ \ \boldsymbol{A}\boldsymbol{d} \preceq \boldsymbol{b}; \quad \boldsymbol{0} \preceq \boldsymbol{d} \preceq \boldsymbol{1}. \tag{5}$$

where the matrix $\boldsymbol{A}$ and $\boldsymbol{b}$ are a matrix and the corresponding right-hand side obtained from the original constraints in (2). $\preceq$ denotes component-wise smaller.

Note that, the parity-check matrix of an (n_s, d_s) SPC product code can be decomposed into d_s sub-matrices such that each sub-matrix satisfies the conditions for Proposition 2, that is, there exists at most 1 in each column of the matrix. Hence, $\boldsymbol{A}$ and $\boldsymbol{b}$ in (5) can be decomposed as $\boldsymbol{A} = \left[\boldsymbol{A}_1^T, ..., \boldsymbol{A}_{d_s}^T\right]^T$ and $\boldsymbol{b} = \left[\boldsymbol{b}_1^T, ..., \boldsymbol{b}_{d_s}^T\right]^T$, and the optimization problem in (5) can be reformulated as

$$\min \quad \boldsymbol{q}^T\boldsymbol{d}, \quad \text{s.t.} \ \ \boldsymbol{A}_1\boldsymbol{d} \preceq \boldsymbol{b}_1; \ ...; \ \boldsymbol{A}_{d_s}\boldsymbol{d} \preceq \boldsymbol{b}_{d_s}; \quad \boldsymbol{0} \preceq \boldsymbol{d} \preceq \boldsymbol{1}. \tag{6}$$

By employing the Lagrange dual method [16], for a given vector $\boldsymbol{p} \succeq \boldsymbol{0}$, (6) can be transformed into the following dual problem

$$L_1(\boldsymbol{p}) \quad = \min_{\boldsymbol{A}_1\boldsymbol{d} \preceq \boldsymbol{b}_1;\ \boldsymbol{0} \preceq \boldsymbol{d} \preceq \boldsymbol{1}} \quad \boldsymbol{q}^T\boldsymbol{d} + \boldsymbol{p}^T(\bar{\boldsymbol{b}}_1 - \bar{\boldsymbol{A}}_1\boldsymbol{d}), \tag{7}$$

where $\bar{\boldsymbol{A}}_1$ consists of all sub-matrices of $\boldsymbol{A}$ other than $\boldsymbol{A}_1$, i.e., $\bar{\boldsymbol{A}}_1 = \boldsymbol{A}/\boldsymbol{A}_1$ and $\bar{\boldsymbol{b}}_1 = \boldsymbol{b}/\boldsymbol{b}_1$. In addition, denote $\boldsymbol{H}_1$ as the parity-check sub-matrix corresponding to $\boldsymbol{A}_1$. Define the auxiliary variables $\boldsymbol{u} = \left[\boldsymbol{u}_1^T, \bar{\boldsymbol{u}_1}^T\right]^T$, where $\boldsymbol{u}_1$ corresponds to the auxiliary variables generated by the parity-check matrix $\boldsymbol{H}_1$. Recall that $\boldsymbol{d} = [\boldsymbol{f}^T, \boldsymbol{u}^T]^T$. Since the sub-matrix of $\bar{\boldsymbol{A}}_1$ corresponding to $\boldsymbol{u}_1$ is a zero matrix, we have $\bar{\boldsymbol{A}}_1\boldsymbol{d} = \hat{\boldsymbol{A}}_1\left[\boldsymbol{f}^T, \bar{\boldsymbol{u}}_1^T\right]^T$, where $\hat{\boldsymbol{A}}$ is the sub-matrix of $\boldsymbol{A}$ corresponding to $\left[\boldsymbol{f}^T, \bar{\boldsymbol{u}}_1^T\right]^T$. Similarly, the sub-matrix of $\boldsymbol{A}_1$ corresponding to $\bar{\boldsymbol{u}}_1$ is a zero matrix, therefore we have $\boldsymbol{A}_1\boldsymbol{d} = \tilde{\boldsymbol{A}}_1\left[\boldsymbol{f}^T, \boldsymbol{u}_1^T\right]^T$, where $\tilde{\boldsymbol{A}}$ is the sub-matrix of $\boldsymbol{A}_1$ corresponding to $\left[\boldsymbol{f}^T, \boldsymbol{u}_1^T\right]^T$. On the other hand, due to Proposition 2, the pseudo-codeword of the LP decoding problem given in (7) must be integer; and thus the constraints $\boldsymbol{A}_1 d \preceq \boldsymbol{b}_1,\ \boldsymbol{0} \preceq \boldsymbol{d} \preceq \boldsymbol{0}$, are equivalent to the following constraints

$$\left[\boldsymbol{H}_1^T\boldsymbol{f}\right]_2 = \boldsymbol{0}, f_i \in \{0,1\}, \forall i;\ \tilde{\boldsymbol{A}}_1\left[\boldsymbol{f}^T, \boldsymbol{u}_1^T\right]^T \preceq \boldsymbol{b}_1,\ \boldsymbol{0} \preceq \boldsymbol{u}_1 \preceq \boldsymbol{1}. \quad (8)$$

Let $\boldsymbol{p}^T\bar{\boldsymbol{A}}_1\boldsymbol{d} = \boldsymbol{p}^T\hat{\boldsymbol{A}}_1\left[\boldsymbol{f}^T, \bar{\boldsymbol{u}}_1^T\right]^T = -\boldsymbol{a}^T\boldsymbol{f} - \bar{\boldsymbol{a}}^T\bar{\boldsymbol{u}}_1$, where $\hat{\boldsymbol{A}}_1^T\boldsymbol{p} = -[\boldsymbol{a}^T, \bar{\boldsymbol{a}}^T]^T$. Denote $\bar{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{a}$, since $\boldsymbol{q}^T\boldsymbol{d} = \boldsymbol{v}^T\boldsymbol{f}$, (7) can be reformulated as

$$\begin{aligned} L_1(\boldsymbol{p}) &= \boldsymbol{p}^T\bar{\boldsymbol{b}}_1 \\ &+ \min_{\left[\boldsymbol{H}_1^T\boldsymbol{f}\right]_2=\boldsymbol{0}, f_i\in\{0,1\},\forall i; \tilde{\boldsymbol{A}}_1\left[\boldsymbol{f}^T,\boldsymbol{u}_1^T\right]^T \preceq \boldsymbol{b}_1, \boldsymbol{0}\preceq \boldsymbol{u}_1 \preceq \boldsymbol{1}} \boldsymbol{v}^T\boldsymbol{f} \\ &\qquad -\boldsymbol{p}^T\hat{\boldsymbol{A}}_1\left[\boldsymbol{f}^T, \bar{\boldsymbol{u}}_1^T\right]^T, \\ &= \boldsymbol{p}^T\bar{\boldsymbol{b}}_1 + \min_{\left[\boldsymbol{H}_1^T\boldsymbol{f}\right]_2=\boldsymbol{0}, f_i\in\{0,1\},\forall i} \bar{\boldsymbol{v}}^T\boldsymbol{f} + \min_{\boldsymbol{0}\preceq\bar{\boldsymbol{u}}_1\preceq\boldsymbol{1}} \bar{\boldsymbol{a}}^T\bar{\boldsymbol{u}}_1. \end{aligned} \quad (9)$$

The second optimization problem can be easily solved by setting the variable as 0 if their coefficients in (9) are positive and set as 1 if their coefficients are negative. In addition, since $\boldsymbol{H}_1$ satisfies Proposition 2, every parity-check of this sub-matrix is independent of the others. Thus the first optimization problem in (9) can be fully decomposed into independent sub-problems as follows. Suppose that there are γ_i ones in the i^{th} row of $\boldsymbol{H}_1$, located at $\{\beta_1, ..., \beta_{\gamma_i}\}$. The sub-problem can be written as

$$\min \sum_{k=1}^{\gamma_i} \bar{\nu}_{\beta_k} f_{\beta_k}, \quad \text{s.t.} \quad \left[\sum_{k=1}^{\gamma_i} f_{\beta_k}\right]_2 = 0,\ f_{\beta_k} \in \{0,1\}. \quad (10)$$

This sub-problem can be efficiently solved by the following Wagner decoding subroutine.

Procedure `Parity-Decoding`($\{\bar{\nu}_{\beta_k}\}_{k=1}^{\gamma_i}$).

- Set $\widehat{f}_{\beta_k} = \frac{-\text{sign}(\bar{\nu}_{\beta_k})+1}{2}$.
  - If $\left[\sum_{k=1}^{\gamma_i} \widehat{f}_{\beta_k}\right]_2 = 0$, stop,
  - otherwise let $\widehat{f}_{\beta_j} \leftarrow 1 - \widehat{f}_{\beta_j}$, where $\beta_j = \text{argmin}(\text{abs}(\bar{\nu}_{\beta_k}))$.
- Output $\{\widehat{f}_{\beta_k}\}_{k=1}^{\gamma_i}$

Assume there are $\bar{m}$ rows in the sub-matrix $\boldsymbol{H}_1$, we can simply run the above sub-routine in parallel for each row to solve the first optimization problem given in (9). After we generate the codeword $\boldsymbol{f}$, we can perform the following heuristic to obtain the auxiliary variables $\boldsymbol{u}$. For the i^{th} row of the matrix $\boldsymbol{H}$, if the obtained solution satisfies this parity-check, we can generate the corresponding auxiliary variables based on the parity-check constraint. If the obtained solution does not satisfy this parity-check constraint, the corresponding auxiliary variables are generated based on their coefficient in (9), i.e., they are set as 0 if their coefficients in (9) are positive and set as 1 if their coefficients are negative.

Assume that the optimal solution to the original Y-LP is $\boldsymbol{d}^*$, we have $L_1(\boldsymbol{p}) \geq \boldsymbol{q}^T\boldsymbol{d}^*$. In order to obtain the optimal solution to the LP problem, we need to solve the following problem

$$\max \quad L_1(\boldsymbol{p}), \quad \text{s.t.} \quad \boldsymbol{p} \succeq \boldsymbol{0}. \quad (11)$$

This problem can be solved by the subgradient method. A subgradient of a concave function $L_1(\cdot)$ at $\boldsymbol{p}$ is a vector $\boldsymbol{h}$ such that $L_1(\boldsymbol{g}) \leq L_1(\boldsymbol{p}) + \boldsymbol{h}^T(\boldsymbol{g} - \boldsymbol{p})$ for all $\boldsymbol{g}$. Denote $\tilde{\boldsymbol{d}}$ as the solution of (9), a subgradient of the function $L_1(\boldsymbol{p})$ is given by $\bar{\boldsymbol{b}}_1 - \bar{\boldsymbol{A}}_1\tilde{\boldsymbol{d}}$. Therefore, we can obtain the following algorithm.

Algorithm 1 *(Decomposition method for Y-LP decoding)*

1) *Set t=0 and* $\boldsymbol{p}^0 = \boldsymbol{0}$.
2) *Solve the primal problem given in (9) as outlined above to obtain* $\boldsymbol{d}^t$.
3) *Calculate the subgradient* $\boldsymbol{s}^t = \boldsymbol{b}_2 - \boldsymbol{A}_2\boldsymbol{d}^t$.
4) *If* $||\boldsymbol{s}^t|| \leq \epsilon$, *stop, otherwise let* $\boldsymbol{p}^{t+1} = [\boldsymbol{p}^t + \theta_t\boldsymbol{s}^t]_+$, $t = t+1$, *go step 2.*

Here $[\cdot]_+$ denote the projection onto the non-negative orthant and ϵ is a pre-defined threshold. θ_t is a stepsize; and if it satisfies certain conditions such as those given in [15], then this method converges.

Furthermore, we may employ a joint subgradient decoder to improve the decoding performance. In this decoder, we can select one group of constraints in (6) as the hard constraints and move the other constraints to the objective function, leading to the following problem that is similar to (7)

$$L_j(\boldsymbol{p}) = \min_{\boldsymbol{A}_j\boldsymbol{d}\preceq\boldsymbol{b}_j; \boldsymbol{0}\preceq\boldsymbol{d}\preceq\boldsymbol{1}} \boldsymbol{q}^T\boldsymbol{d} + \boldsymbol{p}^T(\bar{\boldsymbol{b}}_j - \bar{\boldsymbol{A}}_j\boldsymbol{d}), 1 \leq j \leq d_s. \quad (12)$$

We can run a given number of iterations of the above subgradient decoding algorithm for each problem in (12) and then the one with the maximum objective value is taken as the final result. Such a method can provide a diversity gain and thereby improve the decoding performance. Note that the proposed subgradient decoder is based on a dual-decomposition approach, which is different from the subgradient methods used in [3], [11]. In addition, when the dimension of a product code increases, a multi-level decomposition scheme can be employed to preform the decomposition iteratively, and a corresponding multi-level decomposition decoder can be developed.

Simulation: The convergence behavior of the proposed subgradient decoder for a (4,4) SPC product code is shown in Fig. 1. The performance of the subgradient decoder, the joint subgradient decoder, the LP decoder and the corresponding ML lower bound are illustrated in Fig. 2 for a $(5, 2)$ code. It is seen that the LP decoder can practically achieve the ML lower bound. The performance of the simple subgradient decoder incurs loss of a fractional of a dB with respect to that of the LP decoder. However, the joint subgradient can achieve the LP performance at high SNR at the expense of a higher complexity.

V. Conclusions

We have proved that, for the family of SPC product codes, the fractional distance and pseudo-distance are both equal to the minimum distance. Moreover, we have developed an efficient decoder by exploiting the code structure and our new LP formulation for SPC product codes. To the best of our knowledge, this is the first family of practical codes that have been shown to achieve the ML decoding performance by using methods inspired by LP decoding. Finally we remark that the proposed new LP formulation may open up new avenues to the analysis of other structured block codes as well as the design of corresponding efficient decoding algorithms.

References

[1] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[2] J. Feldman, D. R. Karger, and M. J. Wainwright, "Using linear programming-based decoding of turbo-like codes and its relation to iterative approaches," *Proc. 40th Anu. Allerton Conf. Commun, Contr, & Comput.*, pp. 250–260, Allerton, IL, USA, Oct. 2002.

[3] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, pp. 954–972, Jan. 2005.

[4] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," *Proc. 3rd Int'l. Symp. Turbo Codes & Related Topics*, pp. 75–82, Brest, France, Sept. 2003.

[5] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," *submitted to IEEE Trans. Inform. Theory. Available at* `http://www.arxiv.org/abs/ cs.IT/0512078`.

[6] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," *Proc. Int'l. Symp. Inform. Theory & its Applications*, pp. 991–996, Parma, Italy, Oct. 2004.

[7] P. Chaichanavong and P. H. Siegel, "Relaxation bounds on the minimum pseudo-weight of linear block codes," *Proc. IEEE Int'l Symp. Inform. Theory*, pp. 805–809, Adelaide, Australia, Sept. 2005.

[8] P. O. Vontobel, R. Smarandache, N. Kiyavash, J. Teutsch, and D. Vukobratovic, "On the minimal pseudo-codewords of codes from finite geometries," *Proc. IEEE Int'l Symp. Inform. Theory*, pp. 980-984, Adelaide, Australia, Sept. 2005.

[9] D. M. Rankin, T. A. Gulliver, and D. P. Taylor, "Asymptotic performance of single parity-check product codes ," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2230–2235, Sep. 2003.

[10] K. Yang, J. Feldman, and X. Wang, "Non-linear programming approaches to decoding low-density parity-check codes," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 1603–1613, Aug. 2006.

[11] P. O. Vontobel and R. Koetter, "Towards low-complexity linear-programming decoding," *Proc. 4th Int'l Symp. Turbo Codes & Related Topics*, pp. 1603–1613, Munich, Germany, April 2006.

[12] K. Yang, X. Wang, and J. Feldman, "A New Linear Programming Approach to Decoding Linear Block Codes,"*IEEE Trans. Inform. Theory*, submitted.

[13] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. NewYork, NY: John Wiley, 1998.

[14] D. M. Rankin and T. A. Gulliver, "Single parity-check product codes," *IEEE Trans. Commun.*, vol. 49, pp. 1354–1362, Aug. 2001.

[15] D. Bertsimas and J. N. Tsitsiklis, *Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[16] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 2nd ed., 1999.

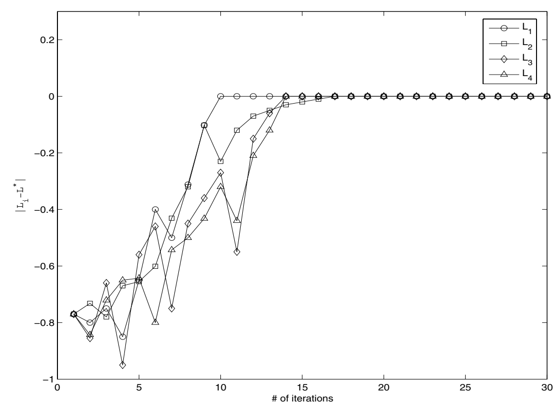

Fig. 1. The convergence behavior of the subgradient decoders against the iteration number for a (4,4) SPC product code. L^* denote the optimal objective value of the problem given in (12). L_j, $1 \leq j \leq 4$, denote the objective values against the iteration number for the j^{th} subgradient decoder.

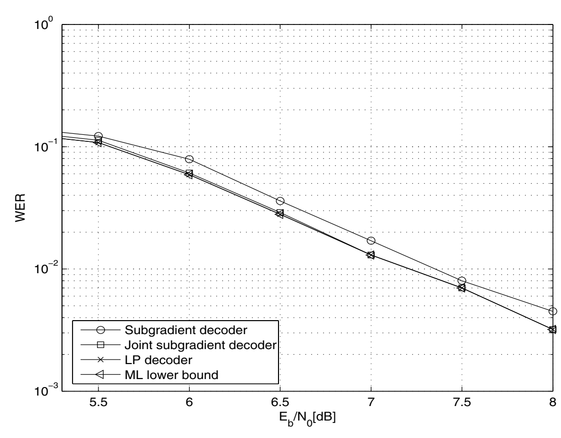

Fig. 2. WER performance of the subgradient decoder, joint subgradient decoder, the LP decoder and the corresponding ML lower bound. A (5,2) SPC product code is used. The code length is 25 and rate is $\left(\frac{4}{5}\right)^2 = 0.64$. The average WER is obtained by counting 100 decoding errors in the simulations.