

BIBLIOTECA

www.inatel.br/biblioteca



**“Este material foi fornecido pela
BIBLIOTECA do Inatel e devido às
restrições de
Direitos Autorais, Lei 9.610/98 que
rege sobre a propriedade
intelectual,
não pode ser distribuído para
outros que não
pertencam à Instituição.”**



www.inatel.br/biblioteca

Linear-Programming Receivers

Mark F. Flanagan

Abstract—It is shown that any communication system which admits a sum-product (SP) receiver also admits a corresponding linear-programming (LP) receiver. The two receivers have a relationship defined by the local structure of the underlying graphical model, and are inhibited by the same phenomenon, which we call *pseudoconfigurations*. This concept is a generalization of the concept of *pseudocodewords* for linear codes. It is proved that the LP receiver has the ‘optimum certificate’ property, and that the receiver output is the lowest cost pseudoconfiguration. Equivalence of graph-cover pseudoconfigurations and linear-programming pseudoconfigurations is also proved. While the LP receiver is generally more complex than the corresponding SP receiver, the LP receiver and its associated pseudoconfiguration structure provide an analytic tool for the analysis of SP receivers. As an example application, we show how the LP design technique may be applied to the problem of joint equalization and decoding.

I. INTRODUCTION

The decoding algorithms for the best known classes of error-correcting code to date, namely concatenated (“turbo”) codes [1] and low-density parity check (LDPC) codes [2], have been shown to be instances of a much more general algorithm called the *sum-product* (SP) algorithm [3], [4], [5]. This algorithm solves the general problem of marginalizing a product of functions which take values in a semiring \mathcal{R} . In the communications context, \mathcal{R} is equal to \mathbb{R} and the maximization of each marginal function minimizes the error rate on a symbol-by-symbol basis. It was also shown that many diverse situations may allow the use of SP based reception [6], including joint iterative equalization and decoding (or *turbo equalization*) [7] and joint source-channel decoding [8].

Recently, a linear-programming (LP) based approach to decoding linear (and especially LDPC) codes was developed for binary [9], [10] and nonbinary coding frameworks [11], [12]. The concept of *pseudocodeword* proved important in the performance analysis of both LP and SP based decoders [13], [14], [15]. Also, linear-programming decoders for irregular repeat-accumulate (IRA) codes and turbo codes were described in [16]. Regarding applications beyond coding, an LP-based method for low-complexity joint equalization and decoding of LDPC coded transmissions over the magnetic recording channel was proposed in [17].

In this paper it is shown that the problem of maximizing a product of \mathcal{R} -valued functions is amenable to an approximate (suboptimal) solution using an LP relaxation, under two conditions: first, that the semiring \mathcal{R} is equal to a subset of

\mathbb{R} , and second, that all non-pendant factor nodes are indicator functions for a local behaviour. Fortunately, these conditions are satisfied by almost all practical communication receiver design problems. Interestingly, the LP exhibits a “separation effect” in the sense that pendant factor nodes in the factor graph contribute the cost function, and non-pendant nodes determine the LP constraint set. This distinction is somewhat analogous to the case of SP-based reception where pendant factor nodes contribute initial messages exactly once, and all other nodes update their messages periodically. Our LP receiver generalizes the LP *decoders* of [10], [12], [16]. It is proved that both the SP and LP based receivers are inhibited by the same phenomenon which we characterize as a set of *pseudoconfigurations*; this is not intuitively obvious since the SP receiver derives from an attempt to minimize error rate on a symbol-by-symbol basis, while the LP receiver derives from an attempt to minimize the configuration error rate.

II. MAXIMIZATION OF A PRODUCT OF FUNCTIONS BY LINEAR PROGRAMMING

We begin by introducing some definitions and notation. Suppose that we have variables x_i , $i \in \mathcal{I}$, where \mathcal{I} is a finite set, and the variable x_i lies in the finite set \mathcal{A}_i for each $i \in \mathcal{I}$. Let $\mathbf{x} = (x_i)_{i \in \mathcal{I}}$; then \mathbf{x} is called a *configuration*, and the Cartesian product $\mathcal{A} = \prod_{i \in \mathcal{I}} \mathcal{A}_i$ is called the *configuration space*. Suppose now that we wish to find that configuration $\mathbf{x} \in \mathcal{A}$ which maximizes the product of real-valued functions

$$u(\mathbf{x}) = \prod_{j \in \mathcal{J}} f_j(\mathbf{x}_j) \quad (1)$$

where \mathcal{J} is a finite set, $\mathbf{x}_j = (x_i)_{i \in \mathcal{I}_j}$ and $\mathcal{I}_j \subseteq \mathcal{I}$ for each $j \in \mathcal{J}$. We define the *optimum* configuration \mathbf{x}_{opt} to be that configuration $\mathbf{x} \in \mathcal{A}$ which maximizes (1). The function $u(\mathbf{x})$ is called the *global function* [5].

The factor graph for the global function $u(\mathbf{x})$ and its factorization (1) is a (bipartite) graph defined as follows. There is a variable node for each variable x_i ($i \in \mathcal{I}$) and a factor node for each factor f_j ($j \in \mathcal{J}$). An edge connects variable node x_i to factor node f_j if and only if x_i is an argument of f_j . Note that for any $j \in \mathcal{J}$, \mathcal{I}_j is the set of $i \in \mathcal{I}$ for which x_i is an argument of f_j . Also, for any $i \in \mathcal{I}$, the set of $j \in \mathcal{J}$ for which x_i is an argument of f_j is denoted \mathcal{J}_i . The *degree* of a node v , denoted $d(v)$, is the number of nodes to which it is joined by an edge. Any node v for which $d(v) = 1$ is said to be *pendant*.

Define

$$\mathcal{Y} = \{i \in \mathcal{I} : \exists j \in \mathcal{J}_i \text{ with } d(f_j) = 1\}$$

M. F. Flanagan was with DEIS, University of Bologna, Via Venezia 52, 47023 Cesena (FC), Cesena, Italy. He is now with Institut für Mathematik, University of Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland (e-mail: mark.flanagan@ieee.org).

i.e., $\mathcal{Y} \subseteq \mathcal{I}$ is the set of $i \in \mathcal{I}$ for which variable node x_i is connected to a pendant factor node. Then, for $i \in \mathcal{Y}$, define

$$h_i(x_i) = \prod_{j \in \mathcal{J}_i : d(f_j)=1} f_j(x_i).$$

We assume that the function $h_i(x_i)$ is positive-valued for each $i \in \mathcal{Y}$. Also, denoting the Cartesian product $\mathcal{A}_{\mathcal{Y}} = \prod_{i \in \mathcal{Y}} \mathcal{A}_i$, we define the projection

$$\mathbf{P}_{\mathcal{Y}} : \mathcal{A} \longrightarrow \mathcal{A}_{\mathcal{Y}} \quad \text{such that} \quad \mathbf{P}_{\mathcal{Y}}(\mathbf{x}) = (x_i)_{i \in \mathcal{Y}}.$$

Also, we adopt the notation $\mathbf{x}_{\mathcal{Y}} = (x_i)_{i \in \mathcal{Y}}$ for elements of $\mathcal{A}_{\mathcal{Y}}$.

Next define $\mathcal{L} = \{j \in \mathcal{J} : d(f_j) \geq 2\}$. So, without loss of generality we may write

$$u(\mathbf{x}) = \prod_{i \in \mathcal{Y}} h_i(x_i) \cdot \prod_{j \in \mathcal{L}} f_j(\mathbf{x}_j) \quad (2)$$

Now, assume that all factor nodes f_j , $j \in \mathcal{L}$, are indicator functions for some local behaviour \mathcal{B}_j , i.e.,

$$f_j(\mathbf{x}_j) = [\mathbf{x}_j \in \mathcal{B}_j] \quad \forall j \in \mathcal{L}$$

where the indicator function for the logical predicate P is defined by

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

Note that we write any $\mathbf{v} \in \mathcal{B}_j$ as $\mathbf{v} = (v_i)_{i \in \mathcal{I}_j}$, i.e., \mathbf{v} is indexed by \mathcal{I}_j . Also we define the *global behaviour* \mathcal{B} as follows: for any $\mathbf{x} \in \mathcal{A}$, we have $\mathbf{x} \in \mathcal{B}$ if and only if $\mathbf{x}_j \in \mathcal{B}_j$ for every $j \in \mathcal{L}$. The configuration $\mathbf{x} \in \mathcal{A}$ is said to be *valid* if and only if $\mathbf{x} \in \mathcal{B}$.

We assume that the mapping $\mathbf{P}_{\mathcal{Y}}$ is injective on \mathcal{B} , i.e., if $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{B}$ and $\mathbf{P}_{\mathcal{Y}}(\mathbf{x}_1) = \mathbf{P}_{\mathcal{Y}}(\mathbf{x}_2)$, then $\mathbf{x}_1 = \mathbf{x}_2$. This corresponds to a ‘well-posed’ problem. Note that in the communications context, since all non-pendant factor nodes are indicator functions, observations may only be contributed through the set of pendant factor nodes. Therefore, failure of the injectivity property in the communications context would mean that one particular set of channel inputs could correspond to two different transmit information sets, which would reflect badly on system design.

So we have

$$\begin{aligned} \mathbf{x}_{\text{opt}} &= \arg \max_{\mathbf{x} \in \mathcal{A}} \left(\prod_{i \in \mathcal{Y}} h_i(x_i) \cdot \prod_{j \in \mathcal{L}} f_j(\mathbf{x}_j) \right) \\ &= \arg \max_{\mathbf{x} : \mathbf{x}_j \in \mathcal{B}_j \forall j \in \mathcal{L}} \prod_{i \in \mathcal{Y}} h_i(x_i) \\ &= \arg \max_{\mathbf{x} \in \mathcal{B}} \sum_{i \in \mathcal{Y}} \log h_i(x_i). \end{aligned}$$

Denote $N_i = |\mathcal{A}_i|$ for $i \in \mathcal{I}$, and $N_{\mathcal{Y}} = \sum_{i \in \mathcal{Y}} N_i$. Now, for each $i \in \mathcal{I}$, define the mapping

$$\xi_i : \mathcal{A}_i \longrightarrow \{0, 1\}^{N_i} \subset \mathbb{R}^{N_i}$$

by

$$\xi_i(\alpha) = ([\gamma = \alpha])_{\gamma \in \mathcal{A}_i}.$$

Building on these mappings, we also define

$$\Xi : \mathcal{A}_{\mathcal{Y}} \longrightarrow \{0, 1\}^{N_{\mathcal{Y}}} \subset \mathbb{R}^{N_{\mathcal{Y}}}$$

according to

$$\Xi(\mathbf{x}_{\mathcal{Y}}) = (\xi_i(x_i))_{i \in \mathcal{Y}}.$$

We note that Ξ is injective.

Now, for vectors $\mathbf{g} \in \mathbb{R}^{N_{\mathcal{Y}}}$, we adopt the notation

$$\mathbf{g} = (\mathbf{g}_i)_{i \in \mathcal{Y}} \quad \text{where} \quad \mathbf{g}_i = (g_i^{(\alpha)})_{\alpha \in \mathcal{A}_i} \quad \forall i \in \mathcal{Y}.$$

In particular, we define the vector $\boldsymbol{\lambda} \in \mathbb{R}^{N_{\mathcal{Y}}}$ by setting

$$\lambda_i^{(\alpha)} = \log h_i(\alpha)$$

for each $i \in \mathcal{Y}$, $\alpha \in \mathcal{A}_i$. This allows us to develop the formulation of the optimum configuration as

$$\begin{aligned} \mathbf{x}_{\text{opt}} &= \arg \max_{\mathbf{x} \in \mathcal{B}} \sum_{i \in \mathcal{Y}} \log h_i(x_i) \\ &= \arg \max_{\mathbf{x} \in \mathcal{B}} \sum_{i \in \mathcal{Y}} \boldsymbol{\lambda}_i \xi_i(x_i)^T \\ &= \arg \max_{\mathbf{x} \in \mathcal{B}} \boldsymbol{\lambda} \Xi(\mathbf{P}_{\mathcal{Y}}(\mathbf{x}))^T. \end{aligned}$$

Note that the optimization has reduced to the maximization of an inner product of vectors, where the first vector derives only from observations (or ‘channel information’) and the second vector derives only from the global behaviour (the set of valid configurations). This problem may then be recast as a linear program

$$\mathbf{x}_{\text{opt}} = \mathbf{P}_{\mathcal{Y}}^{-1}(\Xi^{-1}(\mathbf{g}_{\text{opt}})) \quad (3)$$

where

$$\mathbf{g}_{\text{opt}} = \arg \max_{\mathbf{g} \in \mathcal{K}_{\mathcal{Y}}(\mathcal{B})} \boldsymbol{\lambda} \mathbf{g}^T \quad (4)$$

and the maximization is over the convex hull of all points corresponding to valid configurations:

$$\mathcal{K}_{\mathcal{Y}}(\mathcal{B}) = H_{\text{conv}}\{\Xi(\mathbf{P}_{\mathcal{Y}}(\mathbf{x})) : \mathbf{x} \in \mathcal{B}\}. \quad (5)$$

III. EQUIVALENT LINEAR PROGRAMMING SOLUTION FOR THE OPTIMUM CONFIGURATION

In this section we define a lower-complexity LP to solve for the optimum configuration, and prove that its performance is equivalent to the original. Here, for each $i \in \mathcal{I}$, let α_i be an arbitrary element of \mathcal{A}_i , and let $\mathcal{A}_i^- = \mathcal{A}_i \setminus \{\alpha_i\}$ (note that for each $i \in \mathcal{I}$, $|\mathcal{A}_i| \geq 2$, otherwise x_i is not a ‘variable’). Denote $N_i^- = |\mathcal{A}_i^-| \geq 1$ for every $i \in \mathcal{I}$, and denote $N_{\mathcal{Y}}^- = \sum_{i \in \mathcal{Y}} N_i^-$. Then, for each $i \in \mathcal{I}$, define the mapping

$$\tilde{\xi}_i : \mathcal{A}_i \longrightarrow \{0, 1\}^{N_i^-} \subset \mathbb{R}^{N_i^-}$$

by

$$\tilde{\xi}_i(\alpha) = ([\gamma = \alpha])_{\gamma \in \mathcal{A}_i^-}.$$

Building on this, we also define

$$\tilde{\Xi} : \mathcal{A}_{\mathcal{Y}} \longrightarrow \{0, 1\}^{N_{\mathcal{Y}}^-} \subset \mathbb{R}^{N_{\mathcal{Y}}^-}$$

according to

$$\tilde{\Xi}(\mathbf{x}_{\mathcal{Y}}) = (\tilde{\xi}_i(x_i))_{i \in \mathcal{Y}}.$$

We note that $\tilde{\Xi}$ is injective.

Now, for vectors $\tilde{\mathbf{g}} \in \mathbb{R}^{N_{\mathcal{Y}}^-}$, we adopt the notation

$$\tilde{\mathbf{g}} = (\tilde{g}_i)_{i \in \mathcal{Y}} \quad \text{where} \quad \tilde{g}_i = (\tilde{g}_i^{(\alpha)})_{\alpha \in \mathcal{A}_i^-} \quad \forall i \in \mathcal{Y}.$$

In particular, we define the vector $\tilde{\boldsymbol{\lambda}} \in \mathbb{R}^{N_{\mathcal{Y}}^-}$ by setting

$$\tilde{\lambda}_i^{(\alpha)} = \log \left[\frac{h_i(\alpha)}{h_i(\alpha_i)} \right]$$

for each $i \in \mathcal{Y}$, $\alpha \in \mathcal{A}_i^-$. Our new LP is then given by

$$\mathbf{x}_{\text{opt}} = \mathbf{P}_{\mathcal{Y}}^{-1} \left(\tilde{\Xi}^{-1}(\tilde{\mathbf{g}}_{\text{opt}}) \right) \quad (6)$$

where

$$\tilde{\mathbf{g}}_{\text{opt}} = \arg \max_{\tilde{\mathbf{g}} \in \tilde{\mathcal{K}}_{\mathcal{Y}}(\mathcal{B})} \tilde{\boldsymbol{\lambda}} \tilde{\mathbf{g}}^T \quad (7)$$

and the maximization is over the convex hull of all points corresponding to valid configurations:

$$\tilde{\mathcal{K}}_{\mathcal{Y}}(\mathcal{B}) = H_{\text{conv}} \{ \tilde{\Xi}(\mathbf{P}_{\mathcal{Y}}(\mathbf{x})) : \mathbf{x} \in \mathcal{B} \}. \quad (8)$$

The following proposition proves the equivalence of the original and lower-complexity linear programs.

Proposition 3.1: The linear program defined by (6)–(8) produces the same (optimum) configuration output as the linear program defined by (3)–(5).

Proof: It is easy to show that the simple bijection

$$\mathbf{W} : \tilde{\mathcal{K}}_{\mathcal{Y}}(\mathcal{B}) \longrightarrow \mathcal{K}_{\mathcal{Y}}(\mathcal{B})$$

defined by

$$\mathbf{W}(\tilde{\mathbf{g}}) = \mathbf{g}$$

where

$$\forall i \in \mathcal{Y}, \alpha \in \mathcal{A}_i, \quad g_i^{(\alpha)} = \begin{cases} \tilde{g}_i^{(\alpha)} & \text{if } \alpha \in \mathcal{A}_i^- \\ 1 - \sum_{\beta \in \mathcal{A}_i^-} \tilde{g}_i^{(\beta)} & \text{if } \alpha = \alpha_i \end{cases}$$

which has inverse given by

$$\tilde{g}_i^{(\alpha)} = g_i^{(\alpha)} \quad \forall i \in \mathcal{Y}, \alpha \in \mathcal{A}_i^-$$

has the property that $\tilde{\mathbf{g}} = \tilde{\Xi}(\mathcal{P}_{\mathcal{Y}}(\mathbf{x}))$ if and only if $\mathbf{g} = \mathbf{W}(\tilde{\mathbf{g}}) = \Xi(\mathcal{P}_{\mathcal{Y}}(\mathbf{x}))$. Also observe that if $\mathbf{g} = \mathbf{W}(\tilde{\mathbf{g}})$,

$$\begin{aligned} \tilde{\boldsymbol{\lambda}} \tilde{\mathbf{g}}^T &= \sum_{\alpha \in \mathcal{A}_i^-} [\log h_i(\alpha) - \log h_i(\alpha_i)] \tilde{g}_i^{(\alpha)} \\ &= \sum_{\alpha \in \mathcal{A}_i^-} \log h_i(\alpha) g_i^{(\alpha)} - \log h_i(\alpha_i) [1 - g_i^{(\alpha_i)}] \\ &= \boldsymbol{\lambda} \mathbf{g}^T - \log h_i(\alpha_i) \end{aligned} \quad (9)$$

i.e. the bijection \mathbf{W} preserves the cost function up to an additive constant. Taken together, these two facts imply that $\tilde{\mathbf{g}}$ maximizes $\tilde{\boldsymbol{\lambda}} \tilde{\mathbf{g}}^T$ over $\tilde{\mathcal{K}}_{\mathcal{Y}}(\mathcal{B})$ if and only if $\mathbf{g} = \mathbf{W}(\tilde{\mathbf{g}})$ maximizes $\boldsymbol{\lambda} \mathbf{g}^T$ over $\mathcal{K}_{\mathcal{Y}}(\mathcal{B})$. This proves the result, and justifies the use of the notation \mathbf{x}_{opt} in (6). ■

IV. EFFICIENT LINEAR PROGRAMMING RELAXATION AND ITS PROPERTIES

To reduce complexity of the LP, we introduce auxiliary variables whose constraints, along with those of the elements of $\tilde{\mathbf{g}} \in \mathbb{R}^{N_{\mathcal{Y}}^-}$, will form the relaxed LP problem. We denote these auxiliary variables by

$$p_{j,\mathbf{b}} \quad \text{for each } j \in \mathcal{L}, \mathbf{b} \in \mathcal{B}_j \quad (10)$$

and we form the following vector

$$\mathbf{p} = (\mathbf{p}_j)_{j \in \mathcal{L}} \quad \text{where } \mathbf{p}_j = (p_{j,\mathbf{b}})_{\mathbf{b} \in \mathcal{B}_j} \quad \forall j \in \mathcal{L}.$$

For each $i \in \mathcal{I} \setminus \mathcal{Y}$, let t_i be an arbitrary element of J_i . The constraints of the relaxed LP problem are then

$$\forall j \in \mathcal{L}, \forall \mathbf{b} \in \mathcal{B}_j, \quad p_{j,\mathbf{b}} \geq 0, \quad (11)$$

$$\forall j \in \mathcal{L}, \quad \sum_{\mathbf{b} \in \mathcal{B}_j} p_{j,\mathbf{b}} = 1, \quad (12)$$

$$\begin{aligned} \forall i \in \mathcal{Y}, \forall j \in \mathcal{J}_i \cap \mathcal{L}, \forall \alpha \in \mathcal{A}_i^-, \\ \tilde{g}_i^{(\alpha)} = \sum_{\mathbf{b} \in \mathcal{B}_j, b_i = \alpha} p_{j,\mathbf{b}} \end{aligned} \quad (13)$$

and

$$\begin{aligned} \forall i \in \mathcal{I} \setminus \mathcal{Y}, \forall j \in \mathcal{J}_i \setminus \{t_i\}, \forall \alpha \in \mathcal{A}_i^-, \\ \sum_{\mathbf{b} \in \mathcal{B}_j, b_i = \alpha} p_{j,\mathbf{b}} = \sum_{\mathbf{b} \in \mathcal{B}_{t_i}, b_i = \alpha} p_{t_i,\mathbf{b}}. \end{aligned} \quad (14)$$

Constraints (11)–(14) form a polytope which we denote by $\tilde{\mathcal{Q}}$. The maximization of the objective function $\tilde{\boldsymbol{\lambda}} \tilde{\mathbf{g}}^T$ over $\tilde{\mathcal{Q}}$ forms the relaxed LP problem.

Observe that the further constraints

$$\forall j \in \mathcal{L}, \forall \mathbf{b} \in \mathcal{B}_j, \quad p_{j,\mathbf{b}} \leq 1, \quad (15)$$

$$\forall i \in \mathcal{Y}, \forall \alpha \in \mathcal{A}_i^-, \quad 0 \leq \tilde{g}_i^{(\alpha)} \leq 1 \quad (16)$$

and

$$\forall i \in \mathcal{Y}, \quad \sum_{\alpha \in \mathcal{A}_i^-} \tilde{g}_i^{(\alpha)} \leq 1 \quad (17)$$

follow from the constraints (11)–(14), for any $(\tilde{\mathbf{g}}, \mathbf{p}) \in \tilde{\mathcal{Q}}$.

The receiver algorithm works as follows. First, we say a point in a polytope is *integral* if and only if all of its coordinates are integers. If the LP solution $(\tilde{\mathbf{g}}_{\text{out}}, \mathbf{p})$ is an integral point in $\tilde{\mathcal{Q}}$, the output is the configuration $\mathbf{x}_{\text{out}} = \mathbf{P}_{\mathcal{Y}}^{-1}(\tilde{\Xi}^{-1}(\tilde{\mathbf{g}}_{\text{out}}))$ (we shall prove in the next section that this output is indeed in \mathcal{B}). This configuration may be equal to the optimum configuration (we call this ‘correct reception’) or it may not be (we call this ‘incorrect reception’). Of course, in the communications context, we are usually only interested in a subset of the configuration symbols, namely the information bits. If the LP solution is not integral, the receiver reports a ‘receiver failure’. Note that in this paper, we say that the receiver makes a *reception error* when the receiver output is not equal to the correct configuration (this could correspond to a ‘receiver failure’, or to an ‘incorrect reception’).

V. LP EQUIVALENT TO THE EFFICIENT RELAXATION

We next define another linear program, and prove that its performance is equivalent to that defined in Section IV. This new program is more computationally complex than that defined previously, but (due to its equivalence and simplicity of description) is more useful for theoretical work. For this we define $N = \sum_{i \in \mathcal{I}} N_i$ and

$$\tilde{\Xi} : \mathcal{A} \longrightarrow \{0, 1\}^N \subset \mathbb{R}^N$$

according to

$$\tilde{\Xi}(x) = (\xi_i(x_i))_{i \in \mathcal{I}}.$$

Again, $\tilde{\Xi}$ is injective. For vectors $\tilde{g} \in \mathbb{R}^N$, we denote $\bar{g} = (g_i)_{i \in \mathcal{I}}$ and $g = (g_i)_{i \in \mathcal{Y}}$, where $g_i = (g_i^{(\alpha)})_{\alpha \in \mathcal{A}_i}$ for each $i \in \mathcal{I}$.

The new LP optimizes the cost function λg^T over the polytope \mathcal{Q} defined with respect to variables \bar{g} and p , the constraints (11) and (12), and the following single constraint:

$$\forall j \in \mathcal{L}, \forall i \in \mathcal{I}_j, \forall \alpha \in \mathcal{A}_i, \quad g_i^{(\alpha)} = \sum_{b \in \mathcal{B}_j, b_i = \alpha} p_{j,b}. \quad (18)$$

Note that (15), together with the constraints

$$\forall i \in \mathcal{I}, \forall \alpha \in \mathcal{A}_i, \quad 0 \leq g_i^{(\alpha)} \leq 1 \quad (19)$$

and

$$\forall i \in \mathcal{I}, \quad \sum_{\alpha \in \mathcal{A}_i} g_i^{(\alpha)} = 1 \quad (20)$$

follow from the constraints (11), (12) and (18), for any $(\bar{g}, p) \in \mathcal{Q}$.

In this case, the receiver output is equal to the configuration $x_{\text{out}} = \tilde{\Xi}^{-1}(\bar{g}_{\text{out}})$ in the case where the LP solution $(\bar{g}_{\text{out}}, p)$ is an integral point in \mathcal{Q} (again, this output is in \mathcal{B}), and reports a ‘receiver failure’ if the LP solution is not integral.

The following theorem ensures the equivalence of the two linear programs, and also assures the *optimum certificate* property, i.e., if the receiver output is a configuration, then it is the optimum configuration.

Theorem 5.1: The linear program defined by (11), (12) and (18) produces the same output (configuration or receiver failure) as the linear program defined by (11)–(14). Also, in the case of configuration output, if the receiver output is a configuration, then it is the optimum configuration, i.e., $x_{\text{out}} = x_{\text{opt}}$.

Proof: It is straightforward to show that the mapping

$$\begin{aligned} V : \tilde{\mathcal{Q}} &\longrightarrow \mathcal{Q} \\ (\tilde{g}, p) &\mapsto (\bar{g}, p) \end{aligned}$$

defined by

$$g_i^{(\alpha)} = \begin{cases} \tilde{g}_i^{(\alpha)} & \text{if } i \in \mathcal{Y}, \alpha \in \mathcal{A}_i^- \\ 1 - \sum_{\beta \in \mathcal{A}_i^-} \tilde{g}_i^{(\beta)} & \text{if } i \in \mathcal{Y}, \alpha = \alpha_i \\ \sum_{b \in \mathcal{B}_i, b_i = \alpha} p_{i,b} & \text{if } i \in \mathcal{I} \setminus \mathcal{Y} \end{cases}$$

and with inverse

$$\tilde{g}_i^{(\alpha)} = g_i^{(\alpha)} \quad \forall i \in \mathcal{Y}, \alpha \in \mathcal{A}_i^-$$

is a bijection from one polytope to the other (i.e. $\tilde{g} \in \mathbb{R}^{N_{\tilde{\mathcal{Y}}}}$ satisfies (11)–(14) for some vector p if and only if $\bar{g} \in \mathbb{R}^N$ with $(\bar{g}, p) = V(\tilde{g}, p)$ satisfies (11), (12) and (18) for the same vector p). Also, since $(\bar{g}, p) = V(\tilde{g}, p)$ implies $g = W(\tilde{g})$, (9) implies that the bijection V preserves the cost function up to an additive constant.

Next, we prove that for every configuration $x \in \mathcal{B}$, there exists p such that $(\tilde{\Xi}(x), p) \in \mathcal{Q}$. Let $x \in \mathcal{B}$, and define

$$\forall j \in \mathcal{L}, b \in \mathcal{B}_j, \quad p_{j,b} = \begin{cases} 1 & \text{if } b = (x_i)_{i \in \mathcal{I}_j} \\ 0 & \text{otherwise.} \end{cases}$$

Letting $\tilde{g} = \tilde{\Xi}(\mathcal{P}_{\mathcal{Y}}(x))$ and $\bar{g} = \tilde{\Xi}(x)$, it is easy to check that $(\tilde{g}, p) \in \tilde{\mathcal{Q}}$ and $(\bar{g}, p) \in \mathcal{Q}$ (and that in fact $(\bar{g}, p) = V(\tilde{g}, p)$). This property ensures that every valid configuration $x \in \mathcal{B}$ has a ‘representative’ in the polytope, and thus is a candidate for being output by the receiver.

Next, let $(\tilde{g}, p) \in \tilde{\mathcal{Q}}$ and let $\bar{g} \in \mathbb{R}^N$ be such that $(\bar{g}, p) = V(\tilde{g}, p) \in \mathcal{Q}$. Suppose that all of the coordinates of p are integers. Then, by (11) and (12), for any $j \in \mathcal{L}$ we must have

$$\forall b \in \mathcal{B}_j, \quad p_{j,b} = \begin{cases} 1 & \text{if } b = b^{(j)} \\ 0 & \text{otherwise} \end{cases}$$

for some $b^{(j)} \in \mathcal{B}_j$.

Now we note that for any $i \in \mathcal{I}, j, k \in \mathcal{J}_i \cap \mathcal{L}$, if $b_i^{(j)} = \alpha$ then (using (18))

$$g_i^{(\alpha)} = \sum_{b \in \mathcal{B}_j, b_i = \alpha} p_{j,b} = 1 = \sum_{b \in \mathcal{B}_k, b_i = \alpha} p_{k,b} \quad (21)$$

and thus $b_i^{(j)} = \alpha$. Therefore, there exists $x \in \mathcal{A}$ such that

$$(x_i)_{i \in \mathcal{I}_j} = b^{(j)} \quad \forall j \in \mathcal{L}.$$

Therefore, x is a valid configuration ($x \in \mathcal{B}$). Also we may conclude from (21) that

$$g_i^{(\alpha)} = \begin{cases} 1 & \text{if } x_i = \alpha \\ 0 & \text{otherwise} \end{cases}$$

and therefore $\bar{g} = \tilde{\Xi}(x)$. Also, from the definition of the mapping V , we have $\tilde{g} = \tilde{\Xi}(\mathcal{P}_{\mathcal{Y}}(x))$.

Summarizing these results, we conclude that $(\bar{g}_{\text{opt}}, p) \in \mathcal{Q}$ optimizes the cost function λg^T over \mathcal{Q} and is integral if and only if $(\tilde{g}_{\text{opt}}, p) = V^{-1}(\bar{g}_{\text{opt}}, p) \in \tilde{\mathcal{Q}}$ optimizes the cost function $\lambda \tilde{g}^T$ over $\tilde{\mathcal{Q}}$ and is integral, where $\tilde{\Xi}^{-1}(\bar{g}) = x \in \mathcal{B}$ and $x_{\mathcal{Y}} = \tilde{\Xi}^{-1}(\tilde{g}) = \mathcal{P}_{\mathcal{Y}}(x)$. ■

Thus both LP receivers output either a receiver failure, or the optimum configuration, and have the same performance. The LP receiver of Section IV has lower complexity and is suitable for implementation (e.g. for the program of Section VII); however, for theoretical work the LP of Section V is more suitable (we shall use this polytope throughout Section VI).

VI. PSEUDOCONFIGURATIONS

In this section, we prove a connection between the failure of the LP and SP receivers based on *pseudoconfiguration* concepts.

A. Linear Programming Pseudoconfigurations

Definition 6.1: A linear-programming pseudoconfiguration (LP pseudoconfiguration) is a point $(\bar{\mathbf{g}}, \mathbf{p})$ in the polytope \mathcal{Q} with rational coordinates.

Note that, since the coefficients of the LP are rational, the LP output must be the LP pseudoconfiguration which minimizes the cost function.

B. Factor Graph Covers and Graph-Cover Pseudoconfigurations

We next define what is meant by a finite cover of a factor graph.

Definition 6.2: Let M be a positive integer, and let $\mathcal{S} = \{1, 2, \dots, M\}$. Let \mathcal{G} be the factor graph corresponding to the global function u and its factorization given in (1). A cover configuration of degree M is a vector $\mathbf{x}^{(M)} = (x_i^{(M)})_{i \in \mathcal{I}}$ where $x_i^{(M)} = (x_{i,l})_{l \in \mathcal{S}} \in \mathcal{A}_i^M$ for each $i \in \mathcal{I}$. Define $u^{(M)}$ as the following function of the cover configuration $\mathbf{x}^{(M)}$ of degree M :

$$u^{(M)}(\mathbf{x}^{(M)}) = \prod_{l \in \mathcal{S}} \prod_{j \in \mathcal{J}} f_j(x_{j,l}) \quad (22)$$

where, for each $j \in \mathcal{J}$, $i \in \mathcal{I}_j$, $\Pi_{j,i}$ is a permutation on the set \mathcal{S} , and for each $j \in \mathcal{J}$, $l \in \mathcal{S}$,

$$x_{j,l} = (x_{i,\Pi_{j,i}(l)})_{i \in \mathcal{I}_j}.$$

A cover of the factor graph \mathcal{G} , of degree M , is a factor graph for the global function $u^{(M)}$ and its factorisation (22). In order to distinguish between different factor node labels, we write (22) as

$$u^{(M)}(\mathbf{x}^{(M)}) = \prod_{l \in \mathcal{S}} \prod_{j \in \mathcal{J}} f_{j,l}(x_{j,l})$$

where $f_{j,l} = f_j$ for each $j \in \mathcal{J}$, $l \in \mathcal{S}$.

It may be seen that a cover graph of degree M is a graph whose vertex set consists of M copies of x_i (labelled $x_{i,l}$) and M copies of f_j (labelled $f_{j,l}$), such that for each $j \in \mathcal{J}$, $i \in \mathcal{I}_j$, the M copies of x_i and the M copies of f_j are connected in a one-to-one fashion determined by the permutations $\{\Pi_{j,i}\}$.

We define the cover behaviour \mathcal{B}_M as follows. The cover configuration $\mathbf{x}^{(M)}$ lies in \mathcal{B}_M if and only if $x_{j,l} \in \mathcal{B}_j$ for each $j \in \mathcal{J}$, $l \in \mathcal{S}$.

For any $M \geq 1$, a graph-cover pseudoconfiguration is a valid cover configuration (i.e. one which lies in the behaviour \mathcal{B}_M).

For any graph-cover pseudoconfiguration, we also define the graph-cover pseudoconfiguration vector $\bar{\mathbf{h}} \in \mathbb{R}^N$ according to

$$\bar{\mathbf{h}} = (\mathbf{h}_i)_{i \in \mathcal{I}} \quad \text{where} \quad \mathbf{h}_i = (h_i^{(\alpha)})_{\alpha \in \mathcal{A}_i} \quad \forall i \in \mathcal{I}$$

and

$$h_i^{(\alpha)} = |\{l \in \mathcal{S} : x_{i,l} = \alpha\}|$$

for each $i \in \mathcal{I}$, $\alpha \in \mathcal{A}_i$. Finally, we define the normalized graph-cover pseudoconfiguration vector $\bar{\mathbf{g}} \in \mathbb{R}^N$ by $\bar{\mathbf{g}} = \bar{\mathbf{h}}/M$.

C. Equivalence between Pseudoconfiguration Concepts

In this section, we show the equivalence between the set of LP pseudoconfigurations and the set of graph-cover pseudoconfigurations. The result is summarized in the following theorem.

Theorem 6.1: There exists an LP pseudoconfiguration $(\bar{\mathbf{g}}, \mathbf{p})$ if and only if there exists a graph-cover pseudoconfiguration with normalized pseudoconfiguration vector $\bar{\mathbf{g}}$.

Proof: Suppose $\mathbf{x}^{(M)}$ is a graph-cover pseudoconfiguration for some cover of degree M of the factor graph, and $\bar{\mathbf{g}}$ is its normalized graph-cover pseudoconfiguration vector. Then,

$$g_i^{(\alpha)} = \frac{1}{M} |\{l \in \mathcal{S} : x_{i,l} = \alpha\}|$$

for all $i \in \mathcal{I}$, $\alpha \in \mathcal{A}_i$. Next define \mathbf{p} according to

$$p_{j,\mathbf{b}} = \frac{1}{M} |\{l \in \mathcal{S} : x_{j,l} = \mathbf{b}\}|$$

for all $j \in \mathcal{L}$, $\mathbf{b} \in \mathcal{B}_j$. Then it is easily seen that (11), (12) and (18) are satisfied, and so $(\bar{\mathbf{g}}, \mathbf{p}) \in \mathcal{Q}$.

To prove the other direction, suppose $(\bar{\mathbf{g}}, \mathbf{p}) \in \mathcal{Q}$. Denote by M the lowest common denominator of the (rational) variables $p_{j,\mathbf{b}}$ for $j \in \mathcal{L}$, $\mathbf{b} \in \mathcal{B}_j$. Define $z_{j,\mathbf{b}} = Mp_{j,\mathbf{b}}$ for $j \in \mathcal{L}$, $\mathbf{b} \in \mathcal{B}_j$; these must all be nonnegative integers. Also define $h_i^{(\alpha)} = Mg_i^{(\alpha)}$ for all $i \in \mathcal{I}$, $\alpha \in \mathcal{A}_i$; these must all be nonnegative integers by (18).

We now construct a cover graph of degree M as follows. Begin with M copies of vertex x_i (labelled $x_{i,l}$) and M copies of vertex f_j (labelled $f_{j,l}$), for $i \in \mathcal{I}$, $j \in \mathcal{J}$. Then proceed as follows:

- Label $h_i^{(\alpha)}$ copies of x_i with the value α , for each $i \in \mathcal{I}$, $\alpha \in \mathcal{A}_i$. By (20), all copies of x_i are labelled.
- Label $z_{j,\mathbf{b}}$ copies of f_j with the value \mathbf{b} , for every $j \in \mathcal{L}$, $\mathbf{b} \in \mathcal{B}_j$. By (12), all copies of f_j are labelled.
- Next, let $T_i^{(\alpha)}$ denote the set of copies of x_i labelled with the value α , for $i \in \mathcal{I}$, $\alpha \in \mathcal{A}_i$. Also, for all $j \in \mathcal{L}$, $i \in \mathcal{I}_j$, $\alpha \in \mathcal{A}_i$, let $R_{i,j}^{(\alpha)}$ denote the set of copies of f_j whose label \mathbf{b} satisfies $b_i = \alpha$. The vertices in $T_i^{(\alpha)}$ and the vertices in $R_{i,j}^{(\alpha)}$ are then connected by edges in an arbitrary one-to-one fashion, for every $j \in \mathcal{L}$, $i \in \mathcal{I}_j$, $\alpha \in \mathcal{A}_i$.

Numerically, this connection is always possible because

$$|T_i^{(\alpha)}| = h_i^{(\alpha)} = \sum_{\mathbf{b} \in \mathcal{C}_j, b_i = \alpha} z_{j,\mathbf{b}} = |R_{i,j}^{(\alpha)}|$$

for every $j \in \mathcal{L}$, $i \in \mathcal{I}_j$, $\alpha \in \mathcal{A}_i$. Here we have used (18). Finally, for each $i \in \mathcal{Y}$ and $j \in \mathcal{J}_i$ satisfying $d(f_j) = 1$, the M copies of x_i are connected to the M copies of f_j in an arbitrary one-to-one fashion. It is easy to check that the resulting graph is a cover graph for the original factorization. Therefore, this vertex labelling yields a graph-cover pseudoconfiguration. ■

VII. EXAMPLE APPLICATION: LP-BASED JOINT EQUALIZATION AND DECODING

In this section we consider an example application where we use the above framework to design an LP receiver for a system using binary coding and binary phase-shift keying (BPSK) modulation over a frequency selective channel (with memory L) with additive white Gaussian noise (AWGN). Information-bearing data are encoded to form codewords of the binary code

$$\mathcal{C} = \{c \in GF(2)^n : c\mathcal{H}^T = 0\}$$

where \mathcal{H} is the code's $m \times n$ parity-check matrix over $GF(2)$. Denote the set of code bit indices and parity-check indices by $\mathcal{U} = \{1, 2, \dots, n\}$ and $\mathcal{V} = \{1, 2, \dots, m\}$ respectively. We factor the indicator function for the code into factors corresponding to each local parity check: for $j \in \mathcal{V}$, define the single-parity-check code over $GF(2)$ by

$$\mathcal{C}_j = \{(c_i)_{i \in \mathcal{U}_j} : \sum_{i \in \mathcal{U}_j} c_i = 0\}$$

where $\mathcal{U}_j \subseteq \mathcal{U}$ is the support of the j -th row of \mathcal{H} for each $j \in \mathcal{V}$, and the summation is over $GF(2)$. Thus $c \in \mathcal{C}$ if and only if $(c_i)_{i \in \mathcal{U}_j} \in \mathcal{C}_j$ for each $j \in \mathcal{V}$. The BPSK modulation mapping \mathcal{M} , which maps from $GF(2)$ to \mathbb{R} , is given by $\mathcal{M}(0) = 1$ and $\mathcal{M}(1) = -1$. The channel has $L + 1$ taps $\{h_0, h_1, \dots, h_L\}$ and the received signal for the communication model may be written as

$$r_i = \sum_{t=0}^L h_t \mathcal{M}(c_{i-t}) + n_i$$

where n_i is a zero-mean complex Gaussian random variable with variance σ^2 .

We adopt a state-space (trellis) representation for the channel, with state space $\mathcal{S} = GF(2)^L$; also let $\mathcal{S}^- = \mathcal{S} \setminus \{\mathbf{0}\}$. The local behaviour (or trellis edge set) for the state-space model, denoted \mathcal{D} , is assumed to be time-invariant, although extension to the case of time-variant channel is straightforward. For simplicity we assume that the initial and final states of the channel are not known at the receiver. For $\mathbf{d} \in \mathcal{D}$, let $\text{ip}(\mathbf{d})$, $\text{op}(\mathbf{d})$, $s^S(\mathbf{d})$ and $s^E(\mathbf{d})$ denote the channel input, output, start state and end state respectively. Thus if we set $\mathcal{D} = GF(2)^{L+1}$ and adopt the notation $\mathbf{d} = (d_0 \ d_1 \ \dots \ d_L) \in \mathcal{D}$, we may have $\text{ip}(\mathbf{d}) = d_0$, $s^S(\mathbf{d}) = (d_1 \ d_2 \ \dots \ d_L)$, $s^E(\mathbf{d}) = (d_0 \ d_1 \ \dots \ d_{L-1})$, and $\text{op}(\mathbf{d}) = \sum_{t=0}^L h_t \mathcal{M}(d_t)$. Also let $\mathcal{D}^- = \mathcal{D} \setminus \{\mathbf{0}\}$.

The structure of the LP may be written as follows (we use the efficient LP defined in Section IV). The variables of the LP are ¹

$$\begin{aligned} &\tilde{g}_i^{(\mathbf{d})} \text{ for each } i \in \mathcal{U}, \mathbf{d} \in \mathcal{D}^-, \\ &w_{j,\mathbf{b}} \text{ for each } j \in \mathcal{V}, \mathbf{b} \in \mathcal{C}_j, \\ &q_{i,\mathbf{d}} \text{ for each } i \in \mathcal{U}, \mathbf{d} \in \mathcal{D}, \end{aligned}$$

¹On a correct reception, the LP variables $\{w_{j,\mathbf{b}}\}$ and $\{q_{i,\mathbf{d}}\}$ serve as indicator functions for the single-parity-check codewords and the channel outputs respectively.

and the LP constraints are (here $\mathcal{U}^- = \mathcal{U} \setminus \{n\}$)

$$\forall j \in \mathcal{V}, \forall \mathbf{b} \in \mathcal{C}_j, \quad w_{j,\mathbf{b}} \geq 0 \quad (23)$$

and

$$\forall i \in \mathcal{U}, \forall \mathbf{d} \in \mathcal{D}, \quad q_{i,\mathbf{d}} \geq 0 \quad (24)$$

which follow from (11),

$$\forall j \in \mathcal{V}, \quad \sum_{\mathbf{b} \in \mathcal{C}_j} w_{j,\mathbf{b}} = 1 \quad (25)$$

and

$$\forall i \in \mathcal{U}, \quad \sum_{\mathbf{d} \in \mathcal{D}} q_{i,\mathbf{d}} = 1 \quad (26)$$

from (12),

$$\forall i \in \mathcal{U}, \mathbf{d} \in \mathcal{D}^-, \quad \tilde{g}_i^{(\mathbf{d})} = q_{i,\mathbf{d}} \quad (27)$$

from (13), and

$$\begin{aligned} &\forall i \in \mathcal{U}, \forall j \in \mathcal{U}_j, \\ &\sum_{\mathbf{d} \in \mathcal{D}, \text{ip}(\mathbf{d})=0} q_{i,\mathbf{d}} = \sum_{\mathbf{b} \in \mathcal{C}_j, \mathbf{b}_i=0} w_{j,\mathbf{b}} \end{aligned} \quad (28)$$

and

$$\begin{aligned} &\forall i \in \mathcal{U}^-, \forall s \in \mathcal{S}^-, \\ &\sum_{\mathbf{d} \in \mathcal{D}, s^E(\mathbf{d})=s} q_{i,\mathbf{d}} = \sum_{\mathbf{d} \in \mathcal{D}, s^S(\mathbf{d})=s} q_{i+1,\mathbf{d}} \end{aligned} \quad (29)$$

from (14).

Also, the LP cost function is

$$\sum_{i \in \mathcal{U}} \sum_{\mathbf{d} \in \mathcal{D}^-} \tilde{\lambda}_i^{(\mathbf{d})} \tilde{g}_i^{(\mathbf{d})} \quad (30)$$

where we have, for $i \in \mathcal{U}$, $\mathbf{d} \in \mathcal{D}^-$,

$$\begin{aligned} \tilde{\lambda}_i^{(\mathbf{d})} &= \log \left(\frac{p(r_i | \mathbf{d})}{p(r_i | \mathbf{0})} \right) \\ &= \frac{1}{\sigma^2} \left(|r_i - \text{op}(\mathbf{0})|^2 - |r_i - \text{op}(\mathbf{d})|^2 \right). \end{aligned} \quad (31)$$

Note that in this application, the variables $\tilde{g}_i^{(\mathbf{d})}$, together with the constraint (27), would be removed due to their redundancy, and the variables $q_{i,\mathbf{d}}$ used directly in the cost function (30). The resulting LP is capable of joint equalization and decoding, and has strong links (via Theorems 5.1 and 6.1) to the corresponding ‘‘turbo equalizer’’ based on application of the sum-product algorithm to the same factorization of the global function.

ACKNOWLEDGMENT

The author would like to acknowledge the support of the Institute of Advanced Studies, University of Bologna (ISA-ESRF Fellowship).

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: turbo-codes," *Proc. IEEE International Conference on Communications (ICC)*, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [2] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. IT-8, pp. 21-28, Jan. 1962.
- [3] N. Wiberg, *Codes and Decoding on General Graphs*. Ph.D. Thesis, Linköping University, Sweden, 1996.
- [4] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 325-343, March 2000.
- [5] 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, pp. 498-519, Feb. 2001.
- [6] A. P. Worthen and W. E. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 843-849, Feb. 2001.
- [7] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. Telecomm.*, 6:507-511, Sept.-Oct. 1995.
- [8] N. Görtz, "On the iterative approximation of optimal joint source-channel decoding," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, pp. 1662-1670, Sept. 2001.
- [9] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming*. Ph.D. Thesis, Massachusetts Institute of Technology, Sep. 2003.
- [10] 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, pp. 954-972, March 2005.
- [11] M. F. Flanagan, V. Skachek, E. Byrne, and M. Greferath, "Linear-programming decoding of nonbinary linear codes," *Proc. 7-th International Conference on Source and Channel Coding (SCC 2008)*, Ulm, Germany, Jan. 2008. Arxiv report arXiv:cs.IT/0707.4360v2, October 2007.
- [12] M. F. Flanagan, V. Skachek, E. Byrne, and M. Greferath, "Linear-programming decoding of nonbinary linear codes," submitted to *IEEE Transactions on Information Theory*, Arxiv report arXiv:cs.IT/0804.4384v1, April 2008.
- [13] G. D. Forney, R. Koetter, F. R. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," vol. 123 of *Codes, Systems, and Graphical Models*, IMA Vol. Math. Appl., ch. 5, pp. 101-112, Springer, 2001.
- [14] R. Koetter, W.-C. W. Li, P. O. Vontobel, and J. L. Walker, "Characterizations of pseudo-codewords of LDPC codes," Arxiv report arXiv:cs.IT/0508049, Aug. 2005.
- [15] P. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," to appear in *IEEE Transactions on Information Theory*, Arxiv report arXiv:cs.IT/0512078, Dec. 2005.
- [16] J. Feldman and D. R. Karger, "Decoding turbo-like codes via linear programming," *Proc. 43-rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, pp. 251-260, Nov. 2002.
- [17] M. H. Taghavi and P. H. Siegel, "Graph-based decoding in the presence of ISI," submitted to *IEEE Transactions on Information Theory*, Arxiv report arXiv:cs.IT/0707.1241v1, July 2007.