TOWARDS BRIDGING THE GAP BETWEEN THEORY AND PRACTICE FOR THE SLEPIAN-WOLF PROBLEM

Todd P. Coleman, Muriel Médard

Massachusetts Institute of Technology
Department of EECS
77 Massachusetts Ave, 32-D626
Cambridge, MA 02139

Michelle Effros

California Institute of Technology
Department of Electrical Engineering
1200 East California Blvd, MS 136-93
Pasadena, CA 91125

ABSTRACT

We address practical coding schemes for the Slepian-Wolf distributed data compression problem. We consider three approaches. First, we apply a source-splitting technique to code at any rate in the achievable rate region with low complexity. It is well known that vertices in the achievable rate region can be implemented with low complexity. The source-splitting approach transforms any achievable rate point into a vertex in a higher-dimensional Slepian-Wolf achievable rate region. Secondly, we consider linear programming relaxations of the maximum-likelihood decoding problem. We give a polynomial complexity construction for linear codes with a certificate property. Lastly, when the decoder does not have any knowledge of the source statistics, we present practical schemes for universal decoding, a topic heretofore confined primarily to theory.

1. INTRODUCTION

Distributed compression of correlated sources has become of interest in the research community recently because of its possible promise in efficient transmission of information where energy, computation, and communication constraints prohibit nodes from significantly cooperating with one another. The Slepian-Wolf problem discusses near-lossless distributed compression and has served as a substructure in a number of distributed data dissemination strategies. For discrete memoryless source (DMS) M-tuple $(U^1, \ldots, U^M) \sim P\left(u^1, \ldots, u^M\right)$, the achievable rate region $\mathcal{R}\left[P\left(u^1, \ldots, u^M\right)\right]$ is given [1] by

$$\sum_{i \in S} R_i > H\left(U(S)|U(S^c)\right) \; \forall \; S \subseteq \{1, 2, \ldots, M\} \qquad (1)$$

where $U(S) = \{U^j\}_{j \in S}$.

Csiszár showed in [2] that random linear block codes asymptotically achieve optimal performance - in terms of $\mathcal{R}\left[P\left(u^1, \ldots, u^M\right)\right]$ *and* error exponents. Practically speaking, this problem is difficult because of the complexity in jointly decoding the M sources. However, when operating at *vertices* - rates $(R_1, \ldots, R_M)$ obtained by expanding $H(U^1, \ldots, U^M)$ into M terms by successive applications of the chain rule - a successive decoder given side information of previously decoded users suffices. Recent low-density parity-check code (LDPC) and turbo code formulations have successfully addressed iterative decoding at vertices.

Section 2 provides preliminary definitions. Section 3 discusses *'source-splitting'* to transform *any* achievable rate tuple into a vertex in a higher-dimensional problem. Section 4 considers maximum-likelihood (ML) decoding in terms of a linear program (LP) and discusses an LP relaxation with a certificate property. [2] discussed a *universal* decoder that can achieve the optimal performance without knowing $P\left(u^1, \ldots, u^M\right)$. However, it is NP-hard and is usually only discussed for proofs of existence. Section 5 discusses a low-complexity universal relaxation with a certificate property.

2. PRELIMINARIES

Throughout this discussion we consider a discrete memoryless source (DMS) pair $(U^1, U^2) \in \mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2$ with joint probability distribution $\Pr(u)$ where $u = (u^1, u^2)$. We adhere to the following definitions:

$$\begin{aligned}
CH(\mathcal{S}) &= \text{the convex hull of all } s \in \mathcal{S} \\
\mathcal{V}(\mathcal{B}) &= \{v \in \mathcal{B} \mid v \text{ is a vertex of the polytope } \mathcal{B}\} \\
\mathcal{H}(\mathcal{B}) &\triangleq \text{the number of half-spaces representing } \mathcal{B} \\
\mathcal{P}(\mathcal{U}) &= \left\{ P = \left(\{P_a\}_{a \in \mathcal{U}}\right) : P \geq \underline{0}, \sum_{a \in \mathcal{U}} P_a = 1 \right\} \\
P_{\underline{u}} &= \left(\left\{ \frac{1}{n} \sum_{i=1}^{n} 1_{u_i = a} \right\}_{a \in \mathcal{U}} \right) \text{ for } \underline{u} \in \mathcal{U}^n \\
\mathcal{P}_n(\mathcal{U}) &= \left\{ P \in \mathcal{P}(\mathcal{U}) : P = P_{\underline{u}} \text{ for some } \underline{u} \in \mathcal{U}^n \right\}
\end{aligned}$$

The 'method of types' [2] exploits the property

$$|\mathcal{P}_n(\mathcal{U})| = \binom{n + |\mathcal{U}| - 1}{|\mathcal{U}| - 1} = O\left((n+1)^{|\mathcal{U}|}\right) \qquad (2)$$

to illustrate that *the number of types is polynomial in n*. More generally, we exploit the following repeatedly:

$$\binom{n}{k} = \binom{n}{n-k} \quad = \quad O(n^k). \qquad (3)$$

Here we consider the case where $r \in \{1,2\}$, $|\mathcal{U}_r| = 2^{t_r}$ and block compression transforms $\underline{u}^r \in \mathcal{U}_r^n$ to $\underline{s}^r \in \mathcal{U}_r^{m_r}$ via a linear code $H^r = \begin{bmatrix} -H_1^{r\prime}- \\ \vdots \\ -H_{m_r}^{r\ \prime}- \end{bmatrix} \in \mathcal{U}_r^{m_r \times n}$ according to $\underline{s}^r = H^r \underline{u}^r$ where algebraic operations are performed over $\mathbb{F}_{2^{t_r}}$. For $j \in \{1,\ldots,m_r\}$ we define $N(j) \triangleq \{i | H_{j,i} = 1\}$ and $\delta_j = |N(j)|$. Throughout this discussion we consider achievable rates, as given by (1).

3. SOURCE-SPLITTING

Let us now consider taking each symbol of the DMS U^r and splitting it into a collection of random variables of smaller cardinality. We say that $U_i^r \leftrightarrow (U_i^{r,a}, U_i^{r,b})$ if there is a bijection between the random variables U_i^r and $(U_i^{r,a}, U_i^{r,b})$. We consider the following way to perform source-splitting:

$$\begin{aligned} U_i^r \quad &\mapsto \quad \begin{pmatrix} U_i^{ra} = \min(\pi(U_i^r), T) \\ U_i^{rb} = \max(\pi(U_i^r), T) - T \end{pmatrix} && (4a) \\ &\mapsto \quad U_i = \pi^{-1}\left(U_i^a + U_i^b\right) && (4b) \end{aligned}$$

where $T \in \mathcal{U}_r$ and π is a permutation of $\mathcal{U}_r$.

We note that definition (4) gives many possible splits, especially when we π and $T \in \mathcal{U}_r$. For a nontrivial T there are $\binom{|\mathcal{U}_r|}{T}$ distinct ways to map the $|\mathcal{U}_r|$ symbols to the splitting sets in (4). This provides a total of

$$\sum_{i=1}^{|\mathcal{U}_r|-2} \binom{|\mathcal{U}_r|}{i} = 2^{|\mathcal{U}_r|} - |\mathcal{U}_r| - 2 = O(2^{|\mathcal{U}_r|})$$

distinct ways to perform the splitting mechanism and form the bijection $U_i \leftrightarrow (U_i^a, U_i^b)$.

For two DMSs (U^1, U^2) drawn according to $P\left(u^1, u^2\right)$, we can split U^1 to form (U^{1a}, U^{1b}) as shown in (4). This creates three sources that can be separately encoded at rates R_{1a}, R_{1b}, R_2. Because $U \leftrightarrow (U^{1a}, U^{1b})$, we have $H\left(U^1, U^2\right) = H\left(U^{1a}, U^{1b}, U^2\right)$. Through the chain rule for entropy we consider the rates

$$\begin{aligned} R_{1a} &= H\left(U^{1a}\right) && (5a) \\ R_2 &= H\left(U^2 | U^{1a}\right) && (5b) \\ R_{1b} &= H\left(U^{1b} | U^2, U^{1a}\right) && (5c) \\ R_1 &= R_{1a} + R_{1b}. && (5d) \end{aligned}$$

For any nontrivial split, (R_1, R_2) is not a vertex in $\mathcal{R}\left[P\left(u^1, u^2\right)\right]$ but (R_{1a}, R_2, R_{1b}) *is* a vertex in $\mathcal{R}\left[P\left(u^{1a}, u^2, u^{1b}\right)\right]$. This

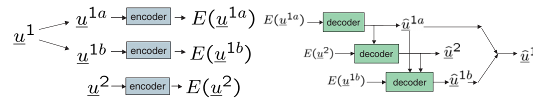

Fig. 1. Source Splitting and Decoding for a Two-Source Slepian-Wolf Problem

directly implies a parallelizable encoding strategy and pipelined single-user decoding strategy that operates with the complexity of a smaller-alphabet decoder. By varying across the different values of the threshold $T \in \mathcal{U}$ and permutation π of $\mathcal{U}_r$, we may sweep across $O(2^{|\mathcal{U}_r|})$ distinct non-vertex points. By treating blocks of outcomes as a single outcome and splitting across the larger alphabet, it can be shown that all rates are achievable with this splitting approach.

3.1. Iterative Decoding and Source Splitting

For rates that are vertices of the Slepian-Wolf region, good binning strategies exist to perform successive decoding. The iterative decoding technique applied here is the sum-product algorithm [3], which operates on the graphical structure of the code. sum-product algorithm produces approximate symbol-wise a posteriori probabilities (APPs). In the context of our problem, the bin indices handed to the decoder for (U^{1a}, U^{1b}, U^2) are denoted as $(\underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2)$. At each level of the pipeline, the APP outputs of previously decoded users are used as inputs to the currently operating decoder. The outputs of the iterative decoders are the approximate APPs

$$\begin{aligned} P\left(U_i^{1a} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^{1a}(u), \\ P\left(U_i^{1b} = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^{1b}(u), \\ P\left(U_i^2 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &\triangleq \mathrm{app}_i^2(u). \end{aligned}$$

Symbol-based Maximum A Posteriori (MAP) allows for $(\hat{\underline{u}}_i^1, \hat{\underline{u}}_i^2)$ the APPs:

$$\hat{u}_i^r = \arg \max_{u \in \{0,1,\ldots|\mathcal{U}_r|-1\}} \mathrm{app}_i^r(i).$$

While $\mathrm{app}_i^2(u)$ is the direct output of one of the iterative decoders, $(\mathrm{app}_i^{1a}(u), \mathrm{app}_i^{1b}(u))$ must be combined to yield $\mathrm{app}_i^1(u)$. The splitting strategy (4) leads to the implication

$$\begin{aligned} j \neq T : U_i^{1a} = j \quad &\Rightarrow \quad U^{1b} = 0 && (6) \\ j \neq 0 : U_i^{1b} = j \quad &\Rightarrow \quad U^{1a} = T && (7) \end{aligned}$$

and thus $\mathrm{app}_i^1(u)$ is easily derived:

$$\begin{aligned} u < T : P\left(U_i^1 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &= \mathrm{app}_i^1(u) \\ u > T : P\left(U_i^1 = u | \underline{s}^{1a}, \underline{s}^{1b}, \underline{s}^2\right) &= \mathrm{app}_i^{1b}(u - T) \end{aligned}$$

Simulation results illustrate the splitting technique's promise. First, $P\left(u^1, u^2\right)$ for sources is randomly selected over $\mathcal{U}_1 =$

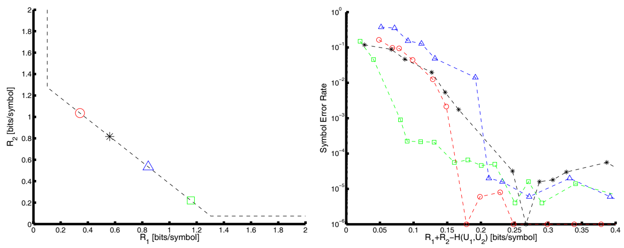

Fig. 2. Symbol error rate for source-splitting to achieve non-vertex rate pairs.

$\mathcal{U}_2 = \mathbb{F}_{2^t}$. We next draw n i.i.d. pairs and encode using irregular LDPCs with degree distributions from [4], where non-zero entries are selected uniformly. We perform the sum-product update rule in its dual form [3, Sec. IX], operating on DFT of APPs to attain the same gain as in the binary case. Figure 2 illustrates the low-complexity decoding at non-vertices for $t = 2, n = 5000$. The leftmost plot shows four non-vertex rate pairs on the boundary of $\mathcal{R}\left[P\left(u^1, u^2\right)\right]$ for which iterative decoding in their neighborhoods has been performed. The rightmost plot shows the symbol error rate as a function of $R_1+R_2-H(U^1,U^2)$. We note from the plots the excellent performance of iterative decoding combined with the proposed splitting technique.

4. LINEAR PROGRAMMING BASED DECODING

The ML decoder uses $\underline{s}$ and $P\left(u^1, u^2\right)$ to select the estimated codeword $\hat{\underline{u}}$ as the most likely from all codewords consistent with $\underline{s}$:

$$\hat{\underline{u}} = \arg \min_{\{\underline{u}^r \in \mathrm{Co}(H^r, \underline{s}^r)\}_{r=1,2}} \sum_{i=1}^{n} -\log P(u_i^1, u_i^2) \quad (8)$$

$$\text{where} \quad \mathrm{Co}\left(H, \underline{s}\right) = \left\{\underline{u} \;\middle|\; H\underline{u} = \underline{s}\right\}.$$

For a linear code, each local constraint is a smaller linear code and

$$\mathrm{Co}\left(H^r, \underline{s}^r\right) = \bigcap_{j=1}^{m_r} \left\{\underline{u} \;\middle|\; \underline{u}_{|N(j)} \in \mathrm{Co}\left(H_j^r, s_j^r\right)\right\}. \quad (9)$$

We now discuss formulating the ML decoding problem in terms of an LP. For $a = (a_1, a_2) \in \mathcal{U}_1 \times \mathcal{U}_2$, we define $I_i^{a_1,a_2}$ to be the indicator variable for the event $(u_i^1, u_i^2) = (a_1, a_2)$. Define

$$\begin{aligned}
\iota^r(I) &= \sum_{a_{\bar{r}} \in \mathcal{U}_{\bar{r}}} I^{a_1,a_2} \text{ where } \bar{r} = \{1,2\} \setminus r \\
\mu^r(I) &= \sum_{a_r \in \mathcal{U}_r} a_r \iota^r(I) \\
\mathcal{I}(H_j^r, s_j^r) &\triangleq \left\{I \;\middle|\; \mu^r(I)_{|N(j)} \in \mathrm{Co}\left(H_j^r, s_j^r\right)\right\} \\
\mathcal{I}(H^r, \underline{s}^r) &= \bigcap_{j=1}^{m_r} \mathcal{I}(H_j^r, s_j^r), \qquad (10)
\end{aligned}$$

and let $\gamma^{a_1,a_2} \triangleq -\log P(a_1, a_2)$ to consider

$$\begin{aligned}
\min \quad & \sum_{r=1}^{2} \sum_{a_r \in \mathcal{U}_r} \sum_{i=1}^{n} \gamma^{a_1,a_2} I_i^{a_1,a_2} \\
s.t. \quad & I \in \mathcal{B} \\
& \mathcal{B} = \{I | \iota^r(I) \in CH(\mathcal{I}(H^r, \underline{s}^r)), r = 1, 2\} \qquad (11)
\end{aligned}$$

By virtue of ML-decoding for linear codes generally being NP-hard, the best bound on $\mathcal{H}\left(\mathcal{B}\right)$ is $O(2^n)$.

Consider an LDPC H over $\mathbb{F}_{2^t}$ such that that for all n, $\forall j, \delta_j \leq d$. We discuss a relaxed polytope in spirit of LP relaxations of Feldman et. al for channel coding [5]. Since $\mathcal{I}(H^r, \underline{s}^r)$ can be represented as (10) we consider

$$\begin{aligned}
\tilde{\mathcal{B}}_j^r(H_j^r, s_j^r) &= \left\{I \mid \mu^r(I)_{|N(j)} \in CH(\mathcal{I}(H_j^r, \underline{s}_j^r))\right\} \\
\tilde{\mathcal{B}} &= \bigcap_{r=1}^{2} \bigcap_{j=1}^{m_r} \tilde{\mathcal{B}}_j^r(H_j^r, s_j^r).
\end{aligned}$$

Because H^r is an LDPC, $\tilde{\mathcal{B}}_j^r(H_j^r, s_j^r)$ can be compactly represented in terms of the $Q^{\delta_j - 1} \leq Q^{d-1}$ legitimate configurations. Thus $\mathcal{H}\left(\tilde{\mathcal{B}}\right) = O(n)$. For any graph other than a tree, however, $\mathcal{V}\left(\tilde{\mathcal{B}}\right)$ includes fractional entries, termed 'pseudocodewords' [6]. Nonetheless it can be shown that

$$v \in \mathcal{V}\left(\tilde{\mathcal{B}}\right) \quad \text{is integral} \;\Rightarrow \{\mu^r(I) \in \mathrm{Co}\left(H^r, s^r\right)\}_{r=1,2}.$$

Thus, as in [5], this relaxation has the ***ML-certificate property***: if an integral solution is found, it is the ML solution.

5. UNIVERSAL MINIMUM-ENTROPY DECODING

In [2], Csiszár discussed a universal decoding algorithm that for linear codes attains all achievable rates and incurs no loss in error exponent. The 'minimum-entropy' decoder selects the coset members with smallest empirical joint entropy:

$$\hat{\underline{u}} = \arg \min_{\{\underline{u}^r \in \mathrm{Co}(H^r, \underline{s}^r)\}_{r=1,2}} H\left(P_{\underline{u}^1, \underline{u}^2}\right). \qquad (12)$$

Note that (12) is a discrete optimization problem with an exponential number of candidates. We now transform it to a continuous optimization problem. For any $I \in \mathcal{V}\left(\mathcal{B}\right)$), and the corresponding $(\underline{u}^1, \underline{u}^2) = (\mu^1(I), \mu^2(I))$, we can construct $P_{\underline{u}^1, \underline{u}^2}$ as a linear mapping:

$$\begin{aligned}
P &= \tau(I), \text{ where} && (13a) \\
P(a_1, a_2) &= \tau_{a_1,a_2}(I) = \frac{1}{n} \sum_{i=1}^{n} I_i^{a_1,a_2}, && (13b)
\end{aligned}$$

Since $H\left(P\right)$ is *strictly concave* in P, and since for concave minimization over a polytope an optimal solution lies in $\mathcal{V}(\mathcal{B})$, we can perform (12) in the continuous domain:

$$\begin{aligned} \min \quad & H(P) & \text{(14a)} \\ s.t. \quad & (I,P) \in \mathcal{B}^{i,p} & \text{(14b)} \\ \text{where} \quad & \mathcal{B}^{i,p} = \{(I,P) | I \in \mathcal{B}, P = \tau(I)\} & \text{(14c)} \end{aligned}$$

and take $(\underline{u}^{1*}, \underline{u}^{2*}) = (\mu^1(I^*), \mu^2(I^*))$ where (I^*, P^*) is an optimal solution to (14).

Along with not knowing how to efficiently represent $\mathcal{B}^{i,p}$, another problem manifests itself in (14): $|\mathcal{V}(\mathcal{B})| = O(2^n)$ and 'concave minimization over a polytope' is NP-hard, generally requiring to visit every $v \in \mathcal{V}(\mathcal{B})$.

Although $|\mathcal{V}(\mathcal{B})| = O(2^n)$, note from (2) that $|\mathcal{P}_n(\mathcal{U})| = O\left((n+1)^{|\mathcal{U}|}\right)$. We thus consider the following strategy:

a) Project $\mathcal{B}^{i,p}$ onto $\mathcal{B}^p = \{P \mid (I,P) \in \mathcal{B}^{i,p} \text{ for some } I\}$.

b) Perform the minimization

$$\begin{aligned} \min \quad & H(P) & \text{(15a)} \\ s.t. \quad & P \in \mathcal{B}^p(H, \underline{s}). & \text{(15b)} \end{aligned}$$

Since $|\mathcal{V}(\mathcal{B}^p)| = O\left((n+1)^{|\mathcal{U}|}\right)$, the worst-case scenario of visiting each $v \in \mathcal{V}(\mathcal{B}^p)$ has polynomial complexity. Let vertex P^* be the minimizer in (15).

c) Find an I^* such that (I^*, P^*) is a vertex of $\mathcal{B}^{i,p}(H, \underline{s})$ and let $\underline{u}^* = \mu(I^*)$ be the estimated codeword.

Performing the the projection of a polytope, as in a), was originally addressed with Fourier-Motzkin elimination [7, section 2.8] and is in general extremely computationally complex. However, in this situation, $\dim(\mathcal{B}^p)$ is fixed and invariant of n, so (2) suggests using special-purpose polytope projection algorithms that are low-complexity in this case. Recent developments [8, Sec. 3], [9] in the optimization literature have illustrated polytope projection algorithms whose complexity is *linear* in $|\mathcal{V}(\mathcal{B}^p)|$ or $\mathcal{H}(\mathcal{B}^p)$. Instantiation of a single LP [7] addresses c).

We consider performing a relaxed universal decoder by performing steps a)-c) replacing

$$\begin{aligned} \mathcal{B}^{i,p} \quad & \text{with} \quad \tilde{\mathcal{B}}^{i,p} = \{(I,P) | I \in \tilde{\mathcal{B}}, P = \tau(I)\}, \text{ and} \\ \mathcal{B}^{p} \quad & \text{with} \quad \tilde{\mathcal{B}}^{p} = \{P \mid (I,P) \in \tilde{\mathcal{B}}^{i,p} \text{ for some } I\}. \end{aligned}$$

Because of the fractional 'pseudocodewords' in $\mathcal{V}\left(\tilde{\mathcal{B}}\right)$, we must verify that $\mathcal{H}\left(\tilde{\mathcal{B}}^p\right)$ and $\left|\mathcal{V}\left(\tilde{\mathcal{B}}^p\right)\right|$ are polynomial in n. Because $\mathcal{H}\left(\tilde{\mathcal{B}}\right) = O(n)$ along with (13), it follows that $\mathcal{H}\left(\tilde{\mathcal{B}}^{i,p}\right) = O(n)$. From [10] and (3) it follows that the projection $\tilde{\mathcal{B}}^p \subseteq \mathbb{R}^{|\mathcal{U}|}$ of a polytope $\tilde{\mathcal{B}}^{i,p}$ satisfies $\mathcal{H}\left(\tilde{\mathcal{B}}^p\right) \leq \binom{\mathcal{H}(\tilde{\mathcal{B}}^{i,p})}{|\mathcal{U}|-1} = O\left(\mathcal{H}\left(\tilde{\mathcal{B}}^{i,p}\right)^{|\mathcal{U}|-1}\right)$. From [7] and (3) it follows that any polytope $\tilde{\mathcal{B}}^p \subseteq \mathbb{R}^{|\mathcal{U}|}$ satisfies $\left|\mathcal{V}\left(\tilde{\mathcal{B}}^p\right)\right| \leq \binom{\mathcal{H}(\tilde{\mathcal{B}}^{p})}{|\mathcal{U}|} = O\left(\mathcal{H}\left(\tilde{\mathcal{B}}^{p}\right)^{|\mathcal{U}|}\right)$. Since $\mathcal{H}\left(\tilde{\mathcal{B}}^{i,p}\right) = O(n)$, both $\mathcal{H}\left(\tilde{\mathcal{B}}^p\right)$ and $\left|\mathcal{V}\left(\tilde{\mathcal{B}}^p\right)\right|$ are polynomial in n. Hence we have constructed a polynomial complexity universal decoder that has the ***ME-certificate property***: if an integral solution is found, it is the minimum-entropy solution.

6. CONCLUSION AND EXTENSIONS

Further work plans to address constructing LDPCs that have provably good properties under the proposed frameworks in Sections 4 and 5.

7. REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[2] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Transactions on Information Theory*, vol. 28, no. 4, pp. 585–592, 1982.

[3] G. D. Forney, "Codes on graphs: Normal realizations," *IEEE Transactions on Information Theory*, pp. 101–112, 2001.

[4] A. Amarou and R. Urbanke, "Ldpcopt," http://lthcwww.epfl.ch/research/ldpcopt/.

[5] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming*, PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, September 2003.

[6] G. Forney, R. Koetter, J. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," *Codes, Systems and graphical models*, pp. 101–112, 2001.

[7] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.

[8] J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat, and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *International Journal of Robotics Research*, vol. 16, no. 1, pp. 11–35, 1997.

[9] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "Equality set projection: A new algorithm for the projection of polytopes in halfspace representation," Tech. Rep. CUED/F-INFENG/TR.463, Cambridge University Engineering Department, March 2004.

[10] G. M. Ziegler, *Lectures on Polytopes*, Springer-Verlag, New York, NY, 1994.