ICIC Express Letters
Part B: Applications
Volume **7**, Number **4**, April **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **753–757**

2-D MULTI-LEVEL MODULATION CODE DESIGN BY INTEGER PROGRAM

Taehyung Park

Department of Industrial and Information Systems Engineering
Soongsil University
No. 369, Sangdo-Ro, Dongjak-Gu, Seoul 156-743, Korea
tpark@ssu.ac.kr

Received September 2015; accepted November 2015

Abstract. *Storing data into a two dimensional pixel image in the holographic data storage (HDS) imposes new constraints in modulation codes. In this paper, we propose a modulation codeword selection procedure based on integer programming models reducing two-dimensional inter-symbol interference (2-D ISI) and inter-page interference (IPI). We formulated an integer programming model that selects codewords satisfying symbol balance and 3-0 violation constraints. We also develop a cutting plane generating sub-model for generating cuts as needed for large scale implementation. We applied the proposed models to 2/3 and 3/4 4-ary codes and compared its performance.*

Keywords: Holographic data storage, Modulation code, Integer programming, Vertex packing

1. **Introduction.** Multi-level holographic data storage is expected to be the next generation optical storage system [1]. Holographic data storage (HDS) systems records information on the volume of holographic materials and writes input data in the form of page. This recording scheme can cause two major problems – two-dimensional (2-D) inter-symbol interference (ISI) and inter-page interference (IPI). 2-D ISI and IPI decrease the bit-error-rate (BER) performance of HDS systems. In a multi-level holographic data storage channel, each pixel records multi-level symbol so that the same number of pixels carry more information than binary pixels. Therefore, there is more severe 2-D ISI between adjacent symbols. To have a good detection performance, many researchers investigated the multi-level modulation codes and error correction codes [2,3].

To reduce 2-D ISI and IPI, the following constraints are usually considered. i) To reduce IPI, the sum of intensity of the signal on each page must be similar. That is, if the distribution of each symbol is equally likely ("balanced"), then the variation caused by the intensity of the signal beam, and IPI could be reduced. ii) To reduce the 2-D ISI, the symbol value difference in adjacent pixels should be small, i.e., symbol '3' is not adjacent to '0'. We call constraint ii) as *3-0 violation*. iii) For the error-correction capability, selected codeword satisfies the minimum Hamming distance. iv) When codewords are stored on a page, the occurrence of the isolated pixel ('0' surrounded with '3' and vice versa) should be minimized [4-6]. For binary code, constant-weight balanced block codes with minimum Hamming distance $d = 2, 4$ are introduced avoiding constraints i) and iii). For example, a 2×2 shaped codeword having two '1's and two '0's is even weight and if all codewords are even weight, the minimum Hamming distance between codewords is at least two. For constraint iv), consecutive 11 or 00 is used to avoid the isolated pixel [7-11].

In this paper, we develop an integer programming model that computes modulation code satisfying constraints i) and ii). The proposed model is a generalized vertex packing problem on a conflict graph. In Section 2, we describe the proposed integer programming model and solution approach. In Section 3, we describe the properties of the 2/3 and 3/4 code computed from the integer program, and Section 4 concludes this paper.

2. Integer Programming Model for Modulation Code.

We describe our model based on the 4-ary modulation code where each codeword is a 3×1 matrix and each cell contains a symbol from $S = \{0, 1, 2, 3\}$. In Figure 1, the first 3×1 matrix shows a typical codeword in 2/3 code. This codeword consists of 2-1-3 symbols. The middle figure shows two 3-0 violations when three codewords are adjacent. 3-0 violation can occur between horizontally and vertically adjacent codewords. The right figure shows an isolated pixel when six codewords are arranged in a page. Here, inside a 3×3 shaded subarea, '0' is surround with '3's.

Suppose that we choose $16 = 4^2$ codewords among possible $64 = 4^3$ codewords. Let $n = 64$, $b = 16$, and $s = 0, \ldots, 3$ denote symbol value in each cell. Let $j = 1, \ldots, n$ denote candidate codeword, c_j the number of 3-0 violation in codeword j, and b_{sj} is the number of symbol s in codeword j. For example, if a codeword consists of 3-0-3, $c_j = 2$ and $b_{3j} = 2$, $b_{0j} = 1$. We define our model on a conflict graph $G = (V, E)$, where $V = \{1, \ldots, n\}$. Edge $(i, j) \in E$, if 3-0 violation occurs when codewords i and j are adjacent. The binary variable $x_j = 1$ if codeword j is selected.

Codeword selection problem is the following integer program.

$$\begin{aligned} \text{Minimize} \quad & \sum_j c_j x_j \\ \text{subject to} \quad & \sum_j x_j = b \\ & \sum_j b_{sj} x_j = d_s, \; s \in S \\ & x_i + x_j \le 1, \; (i,j) \in E \\ & x_j \in \{0,1\}, \; j = 1, \ldots, n. \end{aligned} \tag{1}$$

For Hamming distance constraint, we can similarly define a set $H = \{(i,j) : d_H(i,j) < d_{\min}\}$, where $d_H(i,j)$ is the Hamming distance and $d_{\min}$ is the required minimum Hamming distance. Then we can add constraints $x_i + x_j \le 1$, $(i,j) \in H$. This problem is a variant of a vertex packing problem which is NP-hard [12,13].

When we implement Formula (1) into the integer programming solver such as ILOG CPLEX, for large scale instances, inclusion of edge constraint $x_i + x_j \le 1$, $(i,j) \in E$ into the formulation cannot be feasible. For example, for 6/8 code where $n = 4^8$, $b = 4^6$, the cardinality of the set E is in the order of $O(4^{16})$, so there needs to be an iterative procedure to generate the edge constraints. Also, we need a way to check the violation of edge constraint in a given feasible solution. The next integer programming model decides the given feasible solution contains 3-0 violation when only a subset of edge constraints are included in the current feasible solution. In this formulation, the objective

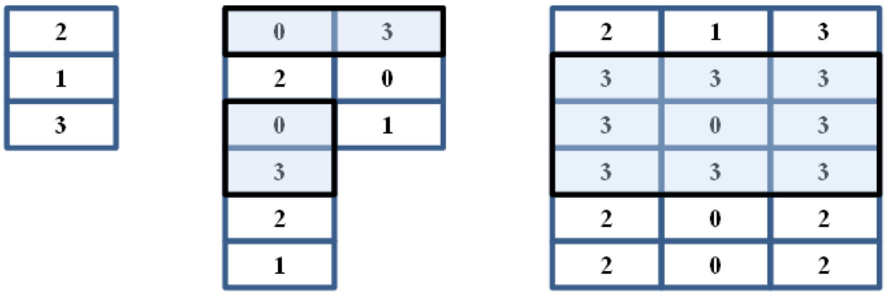

FIGURE 1. Codeword, 3-0 violation, and an isolated pixel

value indicates the number of 3-0 violations in the current solution. Variables $z_{r(i,k),c(i,k)}$ represents symbols of three codewords for checking 3-0 violations such as the middle figure in Figure 1. The y_{kj} represents codeword j is located at kth position. Absolute values s_t computes symbol differences between neighboring codewords horizontally and vertically. The indicator variable $\delta_t = 0$ when the absolute value $|z_{ij}-z_{kl}| \geq l_t$ and if $|z_{ij}-z_{kl}| \leq l_t-1$, $\delta_t = 1$. Here, l_t is an upper bound for the absolute value. We set $l_t=3$ and $M=5$ in (2).

$$\begin{aligned} \text{Minimize} \quad & \sum_t \delta_t \\ \text{subject to} \quad & \sum_j y_{kj} = 1, \ \forall k \\ & z_{r(i,k),c(i,k)} = \sum_j a_{ij} y_{kj}, \ \forall i, k \\ & |z_{ij} - z_{kl}| = s_t, \ t \in A \\ & s_t - M\delta_t \geq l_t, \ t \in A \\ & y_{kj} \text{ and } \delta_t \in \{0,1\}, \ s_t \geq 0 \end{aligned} \tag{2}$$

Note that the optimal value of zero indicates that there exists a 3-0 violation in the current solution and optimal value of four corresponds to that there is no 3-0 violation in the current 2/3 code. The iterative scheme where edge constraints are generated as needed is summarized in Figure 2. Notice that the proposed cutting plane second formulation can be easily modified to cut different isolated pixel patterns.

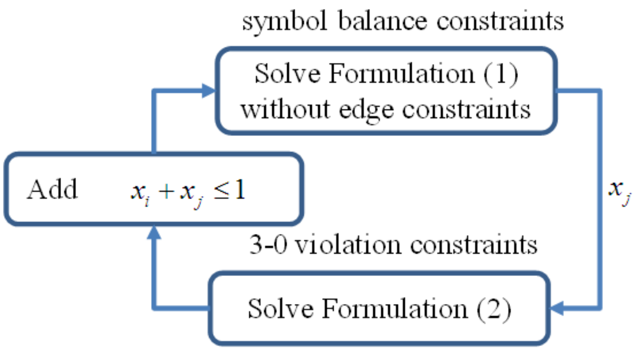
symbol balance constraints
Solve Formulation (1) without edge constraints
x_j
Add $x_i + x_j \leq 1$
3-0 violation constraints
Solve Formulation (2)

Figure 2. Iterative scheme generating cuts

3. **Computational Results.** Formulas (1) and (2) are implemented on PC using Microsoft Visual Studio 10 with IBM ILOG CPLEX as the integer program solver [14]. To speed up the branch and bound procedure in the CPLEX, we added clique constraints to Formula (1). Formula (1) is tested on two types of 4-ary modulation codes. For the 2/3 code, each codeword is 3×1 matrix and we selected 16 codewords among possible 64 codewords. For 3/4 code, each codeword is 2×2 matrix and we selected 64 codewords from 256 possible codewords. In Table 1, we show the selected codewords from Formula (1). For 2/3 codes, symbols 0:1:2:3 are used as 14:14:14:6. Note that for this test, we relaxed symbol balance constraint in (1) using additional slack variables. Also, there are two 3-0 violations inside codewords 031 and 230. There are no 3-0 violations between codewords. Note that all '3' symbols are located in the middle cell, while '0' symbols are in the first and third cells. For 3/4 code, symbols 0:1:2:3 are used as 107:76:73:0. Note that symbol '3' is not used entirely. Because '3' is not used, there is no 3-0 violations inside and between codewords. When we apply penalties for symbol balance variables, we found another code with no '0' symbol that avoids 0-3 violations.

Table 1. Selected codeword set for 2/3 and 3/4 codes

2/3 code	3/4 code			
010	0000	0121	1012	2002
011	0001	0122	1020	2010
012	0002	0200	1021	2011
020	0010	0201	1022	2012
021	0011	0202	1100	2020
022	0012	0210	1101	2021
031	0020	0211	1102	2022
110	0021	0212	1110	2100
120	0022	0220	1120	2101
131	0100	0221	1200	2102
132	0101	0222	1201	2110
210	0102	1000	1202	2120
220	0110	1001	1210	2200
230	0111	1002	1220	2201
231	0112	1010	2000	2202
232	0120	1011	2001	2210

4. **Conclusions.** In this paper, we developed integer programming models for selecting codewords for the holographic storage systems. The modulation code is designed to mitigate ISI and IPI where 3-0 violations between neighboring pixels are avoided and symbols are equally distributed. We also show that 3-0 violation can be checked using separate integer programming model that generates cutting constraint. We tested the proposed models to 4-ary 2/3 and 3/4 codes and computed balanced modulation codes with desired requirements. The performance carried in this paper is purely algebraic cost. To further test the performance of these codes, the solution found in this paper should be tested in terms of BER in various scenarios. Investigation of the valid inequalities for this model is left as a future research.

Acknowledgment. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] H. Coufal, D. Psaltis and G. T. Sincerbox, *Holographic Data Storage*, Springer-Verlag, Berlin, 2000.
[2] B. Marcus, Modulation codes for holographic recording, in *Holographic Data Storage*, H. Coufal, D. Psaltis and G. T. Sincerbox (eds.), Springer-Verlag, Berlin, 2000.
[3] G. W. Burr, J. Ashley, H. Coufal, R. K. Grygier, J. A. Hoffnagle, C. M. Jefferson and B. Marcus, Modulation coding for pixel-matched holographic data storage, *Opt. Lett.*, vol.22, 1997.
[4] B. M. King and M. A. Neifeld, Sparse modulation coding for increased capacity in volume holographic storage, *Appl. Opt.*, vol.39, pp.6681-6688, 2000.
[5] D. E. Pansatiankul and A. A. Sawchuk, Fixed-length two-dimensional modulation coding for imaging page-oriented optical data storage systems, *Appl. Opt.*, vol.42, pp.275-290, 2003.
[6] J. J. Ashley and B. H. Marcus, Two-dimensional low-pass filtering codes, *IEEE Trans. Commun.*, vol.46, 1998.
[7] J. Kim and J. Lee, Two-dimensional 5:8 modulation code for holographic data storage, *Jpn. J. Appl. Phys.*, vol.48, pp.03A031-1-03A031-4, 2009.
[8] J. Kim, J.-K. Wee and J. Lee, Error correcting 4/6 modulation codes for holographic data storage, *Jpn. J. Appl. Phys.*, vol.49, pp.08KB04-1-08KB04-5, 2010.
[9] J. Kim and J. Lee, Error-correcting 6/8 modulation code for reducing two-dimensional intersymbol interference, *Jpn. J. Appl. Phys.*, vol.50, pp.09MB06-1-09MB06-3, 2011.
[10] G. W. Burr, G. Barking, H. Coufal, J. A. Hoffnagle and C. M. Jefferson, Gray-scale data pages for digital holographic data storage, *Opt. Lett.*, vol.23, pp.1218-1220, 1998.

[11] B. M. King, G. W. Burr and M. A. Neifeld, Experimental demonstration of gray-scale sparse modulation codes in volume holographic storage, *Appl. Opt.*, vol.42, pp.2546-2559, 2003.

[12] E. M. Macambira and C. C. de Souza, The edge-weighted clique problem: Valid inequalities, facets and polyhedral computations, *Eur. J. Oper. Res.*, vol.123, pp.346-371, 2000.

[13] H. D. Sherali and J. C. Smith, A polyhedral study of the generalized vertex packing problem, *Math. Prog.*, vol.107, pp.367-390, 2006.

[14] International Business Machines Corporation, *IBM ILOG CPLEX V12.4 User's Manual for CPLEX*, 2011.