Les Cahiers du GERAD ISSN: 0711–2440

Mathematical Programming Formulations for the Design of Convolutional Self-Doubly Orthogonal Codes

B. Jaumard
C. Meyer

G–2006–46

July 2006

Mathematical Programming Formulations for the Design of Convolutional Self-Doubly Orthogonal Codes

Brigitte Jaumard
GERAD and CIISE – Concordia Institute for Information Systems Engineering
Concordia University
1455, boul. de Maisonneuve ouest, CB-410-12
Montréal (Québec) Canada H3G 1M8
bjaumard@CIISE.concordia.ca

Christophe Meyer
GERAD and Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128, Succ. Centre-ville
Montréal (Québec) Canada H3C 3J7
christophe.meyer@gerad.ca

July 2006

Les Cahiers du GERAD
G–2006–46

Abstract

Convolutional Self-Doubly Orthogonal Codes (CSO^{2}C) have been introduced in 1998 by Haccoun et al. as a novel class of convolutional codes which can be decoded using an iterative threshold decoding algorithm that does not require interleavers. However, these codes need to satisfy some orthogonal properties. Moreover, the memory length of the code is a key issue for their overall latency. Unfortunately, the design of CSO^{2}C codes with minimum span corresponds to a highly combinatorial problem and only heuristics have been proposed up to now. We here investigate different mathematical programming formulations for the optimum design of CSO^{2}C codes, or, at least, for deriving a lower bound on their optimum span in order to evaluate the quality of the heuristic solutions. It therefore leads to an assessment on the length of the best known CSO^{2}C codes.

Key Words: Convolutional code, convolutional self-doubly orthogonal code (CSO^{2}C), code span, mixed integer linear program, lower bound.

Résumé

Les codes convolutionnels doublement orthogonaux (CSO^{2}C) ont été introduits en 1998 par Haccoun et al. comme une nouvelle classe de codes convolutionnels pouvant être décodés par un algorithme itératif de décodage à seuil sans entrelaceurs. Ces codes doivent satisfaire certaines propriétés d'orthogonalité et leur longueur est un facteur déterminant de la latence du système. Malheureusement la construction de tels codes de longueur minimale correspond à un problème hautement combinatoire et seules des heuristiques ont été proposées jusqu'à présent. Nous explorons différentes formulations mathématiques pour la construction optimale de codes CSO^{2}C, ou à tout le moins, pour obtenir une borne inférieure sur leur longueur minimale de façon à évaluer la qualité des solutions heuristiques. Ceci conduit à une estimation de la qualité des meilleurs codes connus.

Mots clés : Code convolutionnel, code convolutionnel doublement orthogonal, programme linéaire mixte, borne inférieure.

Acknowledgments: This work was supported by the Concordia University Research Chair on the Optimization of Communication Networks of the first author.

1 Introduction

Haccoun et al. [3, 4] introduced a novel class of convolutional codes, called convolutional Self-Doubly Orthogonal Codes (CSO^{2}C), and studied two types of them: wide-sense convolutional Self-Doubly Orthogonal Codes (CSO^{2}C-WS) and strict-sense convolutional Self-Doubly Orthogonal Codes (CSO^{2}C-SS), see Haccoun et al. [2] for the details. Heuristics have been devised in order to search for CSO^{2}C codes with minimum span for a given number of taps. Both projective geometry methods [2] and pseudorandom computer search [5] have been investigated and provide efficient CSO^{2}C codes. However, no exact method has been devised even only with the goal of computing lower bounds in order to estimate the quality of the heuristic solutions. We therefore study mathematical programming formulations for the design of CSO^{2}C in order to derive lower bounds and hence evaluate the "practical complexity" of finding optimal CSO^{2}C codes, i.e., how much difficult it is in comparison with Golomb rulers for which it is well known that although geometry methods are not exact methods, they do find optimum rulers in practice, while accurate lower bounds are difficult to compute.

The paper is organized as follows. We will restrict our study to CSO^{2}C-WS codes. In Section 2, we provide concise and equivalent definitions of CSO^{2}C-WS codes. In Section 3, we explore various mathematical programming formulations, a straightforward nonlinear one (Subsection 3.1), a first MILP - Mixed Integer Linear Programming - formulation (Subsection 3.2), and then a second MILP formulation (Subsection 3.3) that is more amenable in terms of the number of constraints and variables. We next investigate the linear relaxation of the second MILP formulation in order to derive lower bounds in Section 4. Computational results are summarized in Section 5 and conclusions are drawn in the last section.

2 Definitions

Two different, but equivalent definitions have been given by Cardinal, Haccoun and Gagnon [2] for CSO^{2}C-WS codes. We recall them below and assess their advantages and induced properties.

Definition 1 *A wide-sense convolutional self-doubly orthogonal code (*CSO^{2}C-WS*) of order N is a sequence of N integers $a_1 < a_2 < \cdots < a_N$ such that*

1. *the differences $\delta_{ij} = a_j - a_i$ with $j \neq i$ are distinct;*
2. *the differences of differences $\delta_{k\ell} - \delta_{ij} = (a_\ell - a_k) - (a_j - a_i)$ are distinct for all (i, j, k, ℓ), $\ell \neq k$, $j \neq i$, $k \neq i$, $\ell \neq j$ except for the unavoidable repetitions;*
3. *the differences of differences are distinct from the differences.*

Unavoidable repetitions refer to identities, for example $(a_\ell - a_k) - (a_j - a_i) = (a_\ell - a_j) - (a_k - a_i)$, i.e., differences of differences that are always equal. Since the overall latency of the iterative threshold decoding process is proportional to the memory length of the codes, best codes correspond to those with the smallest memory or, in other words, those with minimum span. Therefore, we are interested in finding the codes of smallest length, i.e., that minimize $a_N - a_1$.

Condition 1) taken alone defines Golomb Ruler with N marks, see, e.g., [11]. Generalization of Golomb Ruler that look like the convolutional self-doubly orthogonal code, but nevertheless with a slightly different definition are studied in [9].

Haccoun, Cardinal and Gagnon [5] claimed that conditions 1) and 3) are implied by condition 2), yielding a shorter definition. To the best of your knowledge, no formal proof of this result is available in the literature (the proof in Baechler's PhD thesis [1] was done for codes corresponding to sets "without the negative", which is a variant of CSO^{2}C codes), therefore we provide a proof of it below.

Proposition 1 *For $N \geq 4$, Conditions 1) and 3) in Definition 1 are implied by Condition 2).*

Proof. Condition 2 states that $(a_\ell - a_k) - (a_j - a_i)$ are distinct for all $(i, j, k, \ell), \ell \neq k, j \neq i, k \neq i, \ell \neq j$. Taking $\ell = i$ and $k = j$, we get that $2(a_i - a_j)$ are distinct for all $(i, j), i \neq j$. This shows that Condition 1 is satisfied.

Let us now show that Condition 3 is also satisfied using a proof by contradiction. Assume that there exists $(i, j, k, \ell, p, q), \ell \neq k, j \neq i, k \neq i, \ell \neq j, p \neq q$ such that

$$(a_\ell - a_k) - (a_j - a_i) = a_q - a_p \tag{1}$$

or equivalently

$$a_i + a_\ell + a_p = a_j + a_k + a_q. \tag{2}$$

Observe first that (1) cannot correspond to unavoidable repetitions. Indeed if it does, then $\{i, \ell, p\} = \{j, k, q\}$. Since $i \neq j, i \neq k$ we have necessarily $i = q$. Similarly $\ell = q$ and $j = k = p$. We then get $2(a_q - a_p) = (a_q - a_p)$, i.e., $a_q = a_p$, a contradiction with the fact that $p \neq q$ and that a CSO^{2}C-WS code is defined by a sequence of distinct integers. Hence (1) does not correspond to unavoidable repetitions.
We distinguish the cases $N \geq 5$ and $N = 4$. Consider first the case $N \geq 5$. Then (1) can be rewritten as:

$$(a_\ell - a_k) - (a_r - a_i) = (a_q - a_p) - (a_r - a_j)$$

with $r \notin \{i, j, \ell, q\}$. If $j \neq p$, this equality contradicts Condition 2, hence (1) does not hold. If $j = p$ but $k \neq p$, we invert the role of j and k. Therefore it remains to consider the case where $j = k = p$. (1) then becomes

$$a_\ell - a_k = a_q - a_i.$$

Since $i \neq k$, this contradicts Condition 1.

If $N = 4$, then some indices in (1) must be identical. Let us first consider the case where one of the indices appearing in the left-hand side of (2) is equal to an index appearing in the right-hand side. Assume for example that $p = j$. Then (1) can be written

$$a_\ell - a_k = a_q - a_i.$$

Since $k \neq i$, this yields a contradiction with Condition 1. The cases $p = k$, $q = i$ and $q = \ell$ are handled similarly. From now on, we can therefore assume $\{i, \ell, p\} \cap \{j, k, q\} = \emptyset$. Moreover the indices in $\{i, \ell, p\}$ play a symmetrical role, as well as the indices in $\{j, k, q\}$. Recall that we must have $|\{i, \ell, p\} \cap \{j, k, q\}| \leq 4$. Exploiting the symmetry, we are left with four cases:

- $i = \ell$ and $j = k$, all other indices being distinct: then (1) can be written

  $$(a_i - a_j) - (a_p - a_i) = (a_j - a_p) - (a_p - a_q).$$

  Since the indices i, j, p, q are all distinct, we get a contradiction with Condition 2.
- $i = \ell = p$, all other indices being distinct. Then (1) can be written

  $$(a_i - a_j) - (a_j - a_i) = (a_k - a_i) - (a_j - a_q)$$

  which is in contradiction with Condition 2.
- $i = \ell = p$, $j = k$, $j \neq q$. In this case, (1) can be written

  $$(a_i - a_q) - (a_r - a_i) = (a_j - a_i) - (a_r - a_j)$$

  where r is chosen such that $r \notin \{i, j, q\}$. Again we get a contradiction with Condition 2.
- $i = \ell = p$ and $j = k = q$. (1) simplifies to $a_i = a_j$, a contradiction.

This shows that Conditions 1 and 3 are implied by Condition 2. □

Note that it is possible to simplify furthermore the Condition 2 of Definition 1, by eliminating some of the unavoidable repetitions. Let introduce the following notation:

$$\delta_{ijk\ell} = a_i + a_\ell - a_j - a_k.$$

Without loss of generality we can assume $i \leq \ell$ and $j \leq k$. Observe furthermore that if a number is present in the set $\{\delta_{ijk\ell} : i \leq \ell, j \leq k\}$, then its opposite is also present. Indeed $\delta_{ijk\ell} = -\delta_{ji\ell k}$ and (j, i, ℓ, k) satisfies the conditions if and only if (i, j, k, ℓ) satisfies them. Therefore Condition 2 can be simplified by adding the condition $\ell > k$, provided that $\delta_{ijk\ell}$ is replaced by $|\delta_{ijk\ell}|$.

We now show that there are no other unavoidable repetitions:

Proposition 2 *Assume that (i,j,k,ℓ) satisfies the conditions $i \neq j, i \neq k, i \leq \ell, j \leq k < \ell$. Then there are no unavoidable repetitions.*

Proof. Assume that for some $(i,j,k,\ell,i',j',k',\ell')$ satisfying $i \neq j, i \neq k, i \leq \ell, j \leq k < \ell, \quad i' \neq j', i' \neq k', i' \leq \ell', j' \leq k' < \ell'$, we have

$$|\delta_{ijk\ell}| = |\delta_{i'j'k'\ell'}|. \tag{3}$$

We will show that we must have $(i,j,k,\ell) = (i',j',k',\ell')$.

There are two cases to distinguish depending on whether or not $\delta_{ijk\ell}$ and $\delta_{i'j'k'\ell'}$ have the same sign or not. In the first case, (3) becomes

$$a_\ell + a_i + a_{k'} + a_{j'} = a_k + a_j + a_{\ell'} + a_{i'}. \tag{4}$$

This holds for all a if and only if (ℓ, i, k', j') is a permutation of (k, j, ℓ', i'). Since $k \neq \ell$, $k \neq i$, $j \neq \ell$ and $j \neq i$, we have necessarily $\{j,k\} = \{j',k'\}$ and $\{i,\ell\} = \{i',\ell'\}$. Since $j \leq k$, $j' \leq k'$, $i \leq \ell$ and $i' \leq \ell'$, we deduce $(i,j,k,\ell) = (i',j',k',\ell')$.

We now consider the case where $\delta_{ijk\ell}$ and $\delta_{i'j'k'\ell'}$ are of opposite sign. (3) becomes

$$a_\ell + a_i + a_{\ell'} + a_{i'} = a_k + a_j + a_{k'} + a_{j'}. \tag{5}$$

For (5) to be true for all a, (ℓ, i, ℓ', i') must be a permutation of (k, j, k', j'). Again since $k \neq \ell$, $k \neq i$, $j \neq \ell$ and $j \neq i$, we have necessarily $\{j,k\} = \{i',\ell'\}$ and $\{j',k'\} = \{i,\ell\}$. Since $j \leq k$, $i' \leq \ell'$, $j' \leq k'$ and $i \leq \ell$, we deduce $(i,j,k,\ell) = (j',i',\ell',k')$. This is not possible because of the conditions $k < \ell$ and $k' < \ell'$.

Hence there are no unavoidable repetitions. □

It follows that Definition 1 can be rewritten:

Definition 2 *A wide-sense convolutional self-doubly orthogonal code (*CSO^{2}C-WS*) of order N is a sequence of N integers $a_1 < a_2 < \cdots < a_N$ such that the $|\delta_{ijk\ell}|$ are distinct for all (i,j,k,ℓ), $i \neq j$, $i \neq k$, $i \leq \ell$, $j \leq k < \ell$.*

Note that the condition defines a partial order on the indices. Completing in all possible ways this partial order to a total order yields the following sets of indices.

$$\begin{aligned} I_1 &= \{(i,j,k,\ell) : 1 \leq i < j \leq k < \ell \leq N\} \\ I_2 &= \{(i,j,k,\ell) : 1 \leq j < i < k < \ell \leq N\} \\ I_3 &= \{(i,j,k,\ell) : 1 \leq j \leq k < i \leq \ell \leq N\}. \end{aligned}$$

Let $I = I_1 \cup I_2 \cup I_3$. Observe that only for $(i, j, k, \ell) \in I_1$, we do not know the sign of $\delta_{ijk\ell}$. Indeed for $(i, j, k, \ell) \in I_2 \cup I_3$, $\delta_{ijk\ell} \geq 0$.
We have

$$\begin{aligned} |I_1| &= \frac{(N-2)(N-1)N(N+1)}{24} \\ |I_2| &= \frac{(N-3)(N-2)(N-1)N}{24} \\ |I_3| &= \frac{(N-1)N(N+1)(N+2)}{24}. \end{aligned}$$

3 Mathematical Formulations for the Optimum Design of CSO^{2}C-WS Codes

3.1 A Compact Nonlinear Integer Formulation

Using the predicate all_different, which is well-known in Constraint Programming (see, e.g., [13]), the problem of finding a CSO^{2}C-WS code with smallest length can be compactly formulated as follows:

$$\min \qquad a_N - a_1$$

subject to:

$$\begin{cases} \delta_{ijkl} = a_i + a_\ell - a_j - a_k & (i, j, k, \ell) \in I \\ \text{all\_different}\,(\{|\delta_{ijk\ell}| : (i, j, k, \ell) \in I\}) & \\ a_{i+1} - a_i \geq 0 & i = 1, \ldots, n-1 \\ a_1 = 0 & \\ a_i \text{ integer} & i = 1, \ldots, n-1. \end{cases}$$

Note that there are two sources of nonlinearities, in addition to the integrality constraint: the all_different constraint, and the absolute values. We next propose two Mixed Integer Linear Programming (MILP) formulations, in which these nonlinearities are removed.

3.2 A First MILP Formulation

Let $\overline{L}$ be an upper bound on the minimum length of a CSO^{2}C-WS code of order N. For instance, set $\overline{L}$ to the length of the best known CSO^{2}C-WS code. Let $D = \{1, 2, \ldots, 2\overline{L}\}$. Then $\delta_{ijk\ell} \in (-D) \cup D$ and $|\delta_{ijk\ell}| \in D$ for all $(i, j, k, \ell) \in I$.

Define

$$\lambda_{ijk\ell u} = \begin{cases} 1 & \text{if } \delta_{ijk\ell} = u \\ 0 & \text{otherwise} \end{cases} \qquad (i, j, k, \ell) \in I, \quad u \in (-D) \cup D.$$

The constraints will enforce that exactly one value of D is assigned to each $|\delta_{ijk\ell}|$. In particular for $(i,j,k,\ell) \in I_1$ and $u \in D$, we will have $\delta_{ijk\ell u} + \delta_{ijk\ell,-u} = 1$ if and only if $|\delta_{ijk\ell}| = u$.

A first MILP formulation, denoted by MILP1, for the CSO^{2}C-WS code problem is as follows:

$$\min \qquad a_N - a_1$$

subject to:

$$a_i + a_\ell - a_j - a_k = \sum_{u \in D} (\lambda_{ijk\ell u} - \lambda_{ijk\ell,-u})u \qquad (i,j,k,\ell) \in I \tag{6}$$

$$\sum_{(i,j,k,\ell) \in I} (\lambda_{ijk\ell,u} + \lambda_{ijk\ell,-u}) \leq 1 \qquad u \in D \tag{7}$$

$$\sum_{u \in D} (\lambda_{ijk\ell u} + \lambda_{ijk\ell,-u}) = 1 \qquad (i,j,k,\ell) \in I \tag{8}$$

$$a_{i+1} - a_i \geq 1 \qquad i = 1, \ldots, N-1 \tag{9}$$

$$a_1 = 0 \tag{10}$$

$$\lambda_{ijk\ell,-u} = 0, \qquad (i,j,k,\ell) \in I \setminus I_1, \quad u \in D \tag{11}$$

$$\lambda_{ijk\ell u}, \lambda_{ijk\ell,-u} \in \{0,1\} \qquad (i,j,k,\ell) \in I, \quad u \in D. \tag{12}$$

This formulation has a very large number of binary variables $\lambda_{ijk\ell,u}$, even if we use (11) to reduce this number. Indeed the number of binary variables is

$$n_B^1 = (2|I_1| + |I_2| + |I_3|)\, 2\overline{L} = \frac{1}{3} N(N-1)(N^2 - N + 1)\overline{L}.$$

By Jaumard and Morel [7], $L^* = \Omega(N^4)$ (where L^* is the minimum length of a CSO^{2}C-WS code of order N), hence $n_B^1 = O(N^8)$.

3.3 A Second MILP Formulation

In this second formulation, the all_different constraint will be on $|\delta_{ijk\ell}|$ rather than on $\delta_{ijk\ell}$. Let $\Delta_{ijk\ell}$ be the variable associated with $|\delta_{ijk\ell}|$. For $(i,j,k,\ell) \in I \setminus I_1$, we have $\Delta_{ijk\ell} = \delta_{ijk\ell}$, so we focus on $(i,j,k,\ell) \in I_1$. Let us introduce the following variables

$$x_{ijk\ell} = \begin{cases} 1 & \text{if } \delta_{ijk\ell} < 0 \\ 0 & \text{otherwise} \end{cases}$$

for all $(i,j,k,\ell) \in I_1$. The following constraints ensure the equality $\Delta_{ijk\ell} = |\delta_{ijk\ell}|$:

$$\begin{aligned}
&\Delta_{ijk\ell} \geq \delta_{ijk\ell} \\
&\Delta_{ijk\ell} \geq -\delta_{ijk\ell} \\
&\Delta_{ijk\ell} \leq \delta_{ijk\ell} + 4\overline{L}x_{ijk\ell} \\
&\Delta_{ijk\ell} \leq -\delta_{ijk\ell} + 4\overline{L}(1 - x_{ijk\ell}).
\end{aligned}$$

We can now deduce a second MILP formulation, denoted by MILP2:

$$\min \qquad a_N - a_1$$

subject to:

$$\Delta_{ijk\ell} = \sum_{u \in D} \mu_{ijk\ell u} u, \qquad (i,j,k,\ell) \in I \tag{13}$$

$$\Delta_{ijk\ell} \geq a_i + a_\ell - a_j - a_k \qquad (i,j,k,\ell) \in I_1 \tag{14}$$

$$\Delta_{ijk\ell} \geq -(a_i + a_\ell - a_j - a_k) \qquad (i,j,k,\ell) \in I_1 \tag{15}$$

$$\Delta_{ijk\ell} \leq a_i + a_\ell - a_j - a_k + 4\overline{L}x_{ijk\ell} \qquad (i,j,k,\ell) \in I_1 \tag{16}$$

$$\Delta_{ijk\ell} \leq -(a_i + a_\ell - a_j - a_k) + 4\overline{L}(1 - x_{ijk\ell}) \qquad (i,j,k,\ell) \in I_1 \tag{17}$$

$$\Delta_{ijk\ell} = a_i + a_\ell - a_j - a_k \qquad (i,j,k,\ell) \in I \setminus I_1 \tag{18}$$

$$\sum_{(i,j,k,\ell) \in I} \mu_{ijk\ell u} \leq 1 \qquad u \in D \tag{19}$$

$$\sum_{u \in D} \mu_{ijk\ell u} = 1 \qquad (i,j,k,\ell) \in I \tag{20}$$

$$a_{i+1} - a_i \geq 1 \qquad i = 1, \ldots, N-1 \tag{21}$$

$$a_1 = 0 \tag{22}$$

$$\mu_{ijk\ell u} \in \{0,1\} \qquad (i,j,k,\ell) \in I, \quad u \in D \tag{23}$$

$$x_{ijk\ell} \in \{0,1\} \qquad (i,j,k,\ell) \in I_1. \tag{24}$$

This formulation has $(2\overline{L} - 1)|I_1|$ less binary variables than the first, but their number is still large:

$$n_B^2 = (|I_1| + |I_2| + |I_3|)\, 2\overline{L} + |I_1|.$$

Table 1 gives the value of the number of binary variables for the two MILP formulations, for some N, assuming $\overline{L}$ equal to the length of the best known CSO^{2}C-WS code.

Table 1: Number of binary variables in the two MILP formulations

N	4	5	6	7	8	9
$\overline{L}$	15	41	100	211	423	807
n_B^1	780	5740	31000	127022	450072	1413864
n_B^2	635	4525	24035	97552	343602	1075134

4 Linear programming relaxation

Denote by (LP1) and (LP2) the linear programming relaxation of the MILP1 and MILP2 formulations respectively. Although we do not know exactly how (LP1) and (LP2) compare, we expect (LP1) to be weaker due to the fact that $\delta_{ijk\ell}$ can be written as convex combination of positive and negative numbers for $(i, j, k, \ell) \in I_1$. Therefore we focus on (LP2).

4.1 Redundant and equivalent constraints

When relaxing constraints (24), constraints (16) and (17) can be removed: indeed given a feasible value for a and Δ, it is always possible to find a value $x_{ijk\ell} \in [0, 1]$ such that these constraints are satisfied for $(i, j, k, \ell) \in I_1$.

Let us explore an alternate way to consider the all_different constraint. Williams and Yan [13] have shown that the convex hull of the feasible solutions of this constraint is described by the inequalities:

$$\sum_{(i,j,k,\ell)\in J} \Delta_{ijk\ell} \geq \sum_{u=1}^{|J|} u \qquad J \subseteq I \tag{25}$$

$$\sum_{(i,j,k,\ell)\in J} \Delta_{ijk\ell} \leq \sum_{u=1}^{|J|} (2\overline{L} + 1 - u) \qquad J \subseteq I. \tag{26}$$

Note that constraints (13), (19), (20) also describe the convex hull of the feasible integer solutions of the all_different constraint when (23) is relaxed. Therefore we replace (13), (19), (20), (23) by (25)-(26). Notice that the constraints (25)-(26) depend only on the variables $\Delta_{ijk\ell}$, but that they are in exponential number.

4.2 Tightening the LP relaxation

Recall that Conditions 1 and 3 of Definition 1 were shown to be implied by Condition 2, see Proposition 1. The LP relaxation can be tightened by reintroducing Conditions 1

and 3. This is done by adding to I the set I_4 defined by:

$$I_4 = \{(i, j, k, \ell) : 1 \leq i = j = k < \ell \leq N\}.$$

The next valid inequalities exploit the observation that a subset of a set of integers $\{a_1, \ldots, a_N\}$ defining a CSO^{2}C-WS code must itself be a CSO^{2}C-WS code. We start with a lemma:

Lemma 1 *Any* CSO^{2}C-WS *code of order* h, $h \geq 2$ *has a length at least equal to*

$$\underline{L}_h = \left\lceil \frac{h(h-1)(h^2-h+6)}{16} \right\rceil. \tag{27}$$

Proof. A proof of this lower bound for $h \geq 4$ is given in Haccoun *et al.* [5, Section IV]. We need to extend this inequality to the cases $h = 2$ and $h = 3$. The inequality (27) is still valid for $h = 2$: indeed the right-hand side becomes 1. For $h = 3$, the right-hand side is 5. In order to extend the validity of the inequality to $h = 3$, we have to show that $a_3 - a_1$ cannot be equal to 3, nor 4 in a CSO^{2}C-WS code defined by $\{a_1, a_2, a_3\}$.

- Case 1: $a_3 - a_1 = 3$. There is only one possible solution up to symmetry: $a_1 = 0$, $a_2 = 1$, $a_3 = 3$. But this solution is not feasible since $2a_2 - 2a_1 = 2 = a_3 - a_2$.
- Case 2: $a_3 - a_1 = 4$. The solution $a = (0, 2, 4)$ is clearly not possible, hence up to symmetry there is again only one solution: $a = (0, 1, 4)$. But this solution violates the constraint $|a_3 - a_2| - |a_2 - a_1| \neq 2(a_2 - a_1)$.

This shows that (27) is valid for $h \geq 2$. □

From Lemma 1, we deduce:

Proposition 3 *The following inequalities are valid:*

$$a_{i+h-1} - a_i \geq \underline{L}_h \qquad i = 1, \ldots, N+1-h, \quad h = 2, \ldots, N \tag{28}$$

$$\Delta_{ijk\ell} \leq a_j + a_\ell - a_i - a_k - 2\underline{L}_h \quad \text{where } h = \min\{\ell - k + 1, j - i + 1\} \qquad (i, j, k, \ell) \in I_1. \tag{29}$$

Proof. Constraints (28) follow from the observation that $\{a_i, a_{i+1}, \ldots, a_{i+h-1}\}$ must be a CSO^{2}C-WS code of order h.
Let us show (29). Note that

$$\Delta_{ijk\ell} = a_j + a_\ell - a_i - a_k - 2\min\{a_j - a_i, a_\ell - a_k\}.$$

From (28), we get

$$\begin{aligned} a_j - a_i &\geq \underline{L}_{j-i+1} \geq \underline{L}_h \\ a_\ell - a_k &\geq \underline{L}_{\ell-k+1} \geq \underline{L}_h. \end{aligned}$$

Hence the result. □

A last class of valid inequalities is given by the following Proposition:

Proposition 4 *The following inequalities hold:*

$$\Delta_{i+1,j,k,\ell} - \Delta_{ijk\ell} \leq a_{i+1} - a_i \qquad 1 \leq i < i+1 < j < k < \ell \leq N \tag{30}$$

$$\Delta_{ijk\ell} - \Delta_{i,j+1,k,\ell} \leq a_{j+1} - a_j \qquad 1 \leq i < j < j+1 < k < \ell \leq N \tag{31}$$

$$\Delta_{ijk\ell} - \Delta_{i,j,k+1,\ell} \leq a_{k+1} - a_k \qquad 1 \leq i < j < k < k+1 < \ell \leq N \tag{32}$$

$$\Delta_{i,j,k,\ell+1} - \Delta_{ijk\ell} \leq a_{\ell+1} - a_\ell \qquad 1 \leq i < j < k < \ell < \ell+1 \leq N. \tag{33}$$

Proof. We give the proof for (33). The proof of the other inequalities is similar. Because of (21), we have $\delta_{ijk,\ell+1} \geq \delta_{ijk\ell}$. There are three cases to consider depending on the position of 0.

(i) $0 \geq \delta_{ijk,\ell+1} \geq \delta_{ijk\ell}$: then $\Delta_{i,j,k,\ell+1} - \Delta_{ijk\ell} = -(\delta_{ijk,\ell+1} - \delta_{ijk\ell}) = -(a_{\ell+1} - a_\ell)$ and (33) is satisfied.

(ii) $\delta_{ijk,\ell+1} \geq 0 \geq \delta_{ijk\ell}$: then $\Delta_{i,j,k,\ell+1} - \Delta_{ijk\ell} = \delta_{ijk,\ell+1} + \delta_{ijk\ell} = 2(a_i - a_j - a_k) + a_{\ell+1} + a_\ell$. Hence $\Delta_{i,j,k,\ell+1} - \Delta_{ijk\ell} - (a_{\ell+1} - a_\ell) = 2(a_i + a_\ell - a_j - a_k) = 2\delta_{ijk\ell} \leq 0$ by assumption. Hence (33) is satisfied.

(iii) $\delta_{ijk,\ell+1} \geq \delta_{ijk\ell} \geq 0$: then $\Delta_{i,j,k,\ell+1} - \Delta_{ijk\ell} = \delta_{i,j,k,\ell+1} - \delta_{ijk\ell} = a_{\ell+1} - a_\ell$, hence (33) is satisfied at equality.

□

4.3 Reducing the number of variables by exploiting the symmetry

Let (LP2+) be the LP relaxation of the MILP2 formulation, modified as explained in Sections 4.1 and 4.2:

$$(LP2+) \quad \min \quad a_N - a_1$$
$$\text{s.t.} \quad \begin{cases} (14) - (15),\ (18) \\ (21) - (22) \\ (25) - (26) \\ (28) - (33). \end{cases}$$

As it is the case for the Golomb Ruler problem [6] or for the more general DTS problem [12], the number of variables can be reduced roughly by an half by exploiting the symmetry. This reduction is based on the following result.

Proposition 5 *There exists an optimal solution to (LP2+) satisfying*

$$a_i = a_N - a_{N-i+1} \qquad i = 1, \ldots, N \tag{34}$$
$$\Delta_{ijk\ell} = \Delta_{N-\ell+1,N-k+1,N-j+1,N-i+1} \qquad (i,j,k,\ell) \in I_1 \tag{35}$$
$$\Delta_{ijk\ell} = \Delta_{N-k+1,N-\ell+1,N-i+1,N-j+1} \qquad (i,j,k,\ell) \in I_2 \cup I_3 \tag{36}$$
$$\Delta_{iii\ell} = \Delta_{N-\ell+1,N-\ell+1,N-\ell+1,N-i+1} \qquad (i,i,i,\ell) \in I_4. \tag{37}$$

Proof. Let (a^*, Δ^*) be an optimal solution. We will show that an alternate optimal solution is $(\tilde{a}, \tilde{\Delta})$ with

$$\tilde{a}_i = a^*_N - a^*_{N-i+1} \qquad i = 1, \ldots, N \tag{38}$$
$$\tilde{\Delta}_{ijk\ell} = \Delta^*_{N-\ell+1,N-k+1,N-j+1,N-i+1} \qquad (i,j,k,\ell) \in I_1 \tag{39}$$
$$\tilde{\Delta}_{ijk\ell} = \Delta^*_{N-k+1,N-\ell+1,N-i+1,N-j+1} \qquad (i,j,k,\ell) \in I_2 \cup I_3 \tag{40}$$
$$\tilde{\Delta}_{iii\ell} = \Delta^*_{N-\ell+1,N-\ell+1,N-\ell+1,N-i+1} \qquad (i,i,i,\ell) \in I_4. \tag{41}$$

Then by linearity, $(\hat{a}, \hat{\Delta}) = \frac{1}{2}(a^*, \Delta^*) + \frac{1}{2}(\tilde{a}, \tilde{\Delta})$ is also an optimal solution. This optimal solution satisfies the condition of the Proposition.
So let us show that $(\tilde{a}, \tilde{\Delta})$ is an optimal solution. Notice that

$$(i,j,k,\ell) \in I_1 \quad \Leftrightarrow \quad (N-\ell+1, N-k+1, N-j+1, N-i+1) \in I_1 \tag{42}$$
$$(i,j,k,\ell) \in I_2 \cup I_3 \quad \Leftrightarrow \quad (N-k+1, N-\ell+1, N-i+1, N-j+1) \in I_2 \cup I_3 \tag{43}$$
$$(i,i,i,\ell) \in I_4 \quad \Leftrightarrow \quad (N-\ell+1, N-\ell+1, N-\ell+1, N-i+1) \in I_4. \tag{44}$$

Assume that $(i,j,k,\ell) \in I_1$ and consider (14).

$$\begin{aligned} \tilde{\Delta}_{ijk\ell} &= \Delta^*_{N-\ell+1,N-k+1,N-j+1,N-i+1} \\ &\geq -\left(a^*_{N-\ell+1} + a^*_{N-i+1} - a^*_{N-k+1} - a^*_{N-j+1}\right) \\ &= \tilde{a}_i + \tilde{a}_\ell - \tilde{a}_j - \tilde{a}_k \end{aligned}$$

where we used (15) and (42). This shows that (14) is satisfied. By inverting the role of (14) and (15) , we show similarly that (15) is satisfied.
Assume that $(i,j,k,\ell) \in I_2 \cup I_3$. Then

$$\begin{aligned} \tilde{\Delta}_{ijk\ell} &= \Delta^*_{N-k+1,N-\ell+1,N-i+1,N-j+1} \\ &= a^*_{N-k+1} + a^*_{N-j+1} - a^*_{N-\ell+1} - a^*_{N-i+1} \\ &= \tilde{a}_i + \tilde{a}_\ell - \tilde{a}_j - \tilde{a}_k \end{aligned}$$

where we used (18) and (43). This shows that (18) is satisfied for $I_2 \cup I_3$. We show in a similar way that (18) is satisfied for I_4.
Assume that $(i,j,k,\ell) \in I_1$ and consider (29). We have

$$\begin{aligned}\tilde{\Delta}_{ijk\ell} &= \Delta^*_{N-\ell+1,N-k+1,N-j+1,N-i+1} \\ &\leq a^*_{N-k+1} + a^*_{N-i+1} - a^*_{N-\ell+1} - a^*_{N-j+1} - 2\underline{L}_{\min\{j-i+1,\ell-k+1\}} \\ &= \tilde{a}_\ell + \tilde{a}_j - \tilde{a}_k - \tilde{a}_i - 2\underline{L}_{\min\{j-i+1,\ell-k+1\}}\end{aligned}$$

which shows that (29) is satisfied.
Consider now (25). Let J be a subset of I. By (42)-(44), $\sum_{(i,j,k,\ell)\in J} \tilde{\Delta}_{ijk\ell}$ is a sum of variables $\Delta^*_{i'j'k'\ell'}$ for a set J' with $|J'| = |J|$. Hence (25) is satisfied.
It can be shown in a similar way that the other constraints are also satisfied. Hence $(\tilde{a}, \tilde{\Delta})$ is a feasible solution to $(LP2+)$. Its objective value is $\tilde{a}_N - \tilde{a}_1 = (a^*_N - a^*_1) - (a^*_N - a^*_N) = a^*_N - a^*_1$, which shows that $(\tilde{a}, \tilde{\Delta})$ is an optimal solution. □

5 Computational results

We solved the linear program (LP2+) with roughly half the variables eliminated using (34)-(37). Due to their exponential number, constraints (25) were generated on a "as needed basis" in a similar way than [10]. More specifically we generated only constraints that are violated by more than 10^{-4} by the current solution. We did not consider (26). Constraints (28) and (30)-(33) did not help, so we removed them.

$\underline{z}^H$ denotes the value of the lower bound reported in [5]; $\overline{z}$ corresponds to the span of the best code found in Jaumard and Morel [7] and Jaumard and Solari [8]; $\underline{z}^{\rm LP}_{1234}$ is the lower bound obtained by solving the LP with sets I_1, I_2, I_3 and I_4 (this corresponds to Definition 1) while $\underline{z}^{\rm LP}_{123}$ is the same lower bound without set I_4 (this corresponds to the simplified Definition 2). The gap is defined as

$$\text{gap} = \frac{\overline{z} - \lceil \underline{z}^{\rm LP}_{1234} \rceil}{\overline{z}}.$$

The computational results are presented in Table 2. A star (*) in the column for $\overline{z}$ indicates that the upper bound is known to be the optimal value (see Jaumard and Solari [8]).[1]

[1] Note that the codes for $N = 5$ and $N = 6$ reported in Table II of [5] are incorrect: indeed for $N = 5$, $|\delta_{1345}| = 28 = |\delta_{5235}|$, and for $N = 6$, $|\delta_{6556}| = 6 = |\delta_{3456}|$.

Table 2: Computational results

N	$\underline{z}^H$	$\underline{z}^{\text{LP}}_{123}$	$\underline{z}^{\text{LP}}_{1234}$	$\overline{z}$	gap (%)
4	13.5	9.84	13.94	15*	6.67
5	32.5	27.14	34.90	41*[2]	14.63
6	67.5	62.23	74.73	100*[2]	25.00
7	126.0	125.76	143.71	211*[8]	31.75
8	217.0	229.77	254.74	423*[8]	39.72
9	351.0	388.68	421.78	807 [8]	47.71
10	540.0	618.49	660.54	1,475 [8]	55.19
11	797.5	940.17	992.51	2,767 [8]	64.11
12	1,138.5	1,374.71	1,438.30	4,988 [8]	71.15
13	1,579.5	1,943.70	2,020.02	8,405 [8]	75.95
14	2,138.5	2,674.77	2,764.71	11,347 [7]	75.63
15	2,835.0	3,599.63	3,704.10	20,792 [7]	82.18
16	3,690.0	4,739.93	4,860.35	25,396 [7]	80.86
17	4,726.0	6,136.25	6,273.79	30,387 [7]	79.35
18	5,967.0	7,823.54	7,979.01	38,426 [7]	79.23
19	7,438.5	9,832.54	10,007.32	53,657 [7]	81.35
20	9,167.5	12,209.18	12,404.15	62,345 [7]	80.10
21	11,183.0	14,994.34	15,210.89	104,310 [7]	85.42
22	13,513.5	18,228.31	18,467.33	116,314 [7]	84.12
23	16,192.0	21,961.06	22,223.88	128,609 [7]	82.72
24	19,251.0	26,242.47	26,530.25	143,280 [7]	81.48
25	22,725.0	31,117.68	31,431.39	198,899 [7]	84.20
26	26,650.0	36,647.99	36,988.55	210,825 [7]	82.46
27	31,063.5	42,882.13	43,251.38	277,146 [7]	84.39
28	36,004.5	49,881.35	50,280.44	301,619 [7]	83.33
29	41,513.5	57,670.79	58,077.46	363,589 [7]	84.03
30	47,632.5	65,493.40	65,932.13	412,259 [7]	84.01

6 Conclusion

We have presented two mixed integer linear formulations for the optimum design of CSO^{2}C-WS codes. Adding valid inequalities to the second formulation allowed us to compute new lower bounds for the optimum span of CSO^{2}C-WS codes, by solving the LP relaxation. These lower bounds improve significantly on the ones proposed by [5]. Comparing these lower bounds with the value of the span of the best known CSO^{2}C-WS codes however shows the existence of a gap, which is much larger than for convolutional orthogonal codes, see

[11]. This clearly shows the need of additional works to both strengthen the lower bound and to design more efficient methods for the derivation of good CSO^{2}C-WS codes.

References

[1] B. Baechler. Analyse et détermination de codes doublement orthogonaux pour décodage itératif. Master's thesis, École Polytechnique de Montreal, 2000.

[2] C. Cardinal, D. Haccoun, and F. Gagnon. Iterative threshold decoding without interleaving for convolutional self-doubly orthogonal codes. *IEEE Transactions on Communications*, 51(8):1274–1282, 2003.

[3] C. Cardinal, D. Haccoun, F. Gagnon, and N. Batani. Convolutional self doubly orthogonal codes for iterative decoding without interleaving. In *Proceedings of the 1998 IEEE International Symposium on Information Theory*, page 280, 1998.

[4] F. Gagnon, D. Haccoun, N. Batani, and C. Cardinal. Apparatus for convolutional self-doubly orthogonal encoding and decoding. U.S. Patent 6,167,225, December 2000.

[5] D. Haccoun, C. Cardinal, and F. Gagnon. Search and determination of convolutional self-doubly orthogonal codes for iterative threshold decoding. *IEEE Transactions on Communications*, 53(5):802–809, 2005.

[6] P. Hansen, B. Jaumard, and C. Meyer. On lower bounds for numbered complete graphs. *Discrete Applied Mathematics*, 94(1-3):205–225, 1999.

[7] B. Jaumard and M. Morel. Enhancing algebraic methods for the design of better cso^{2}c codes. in preparation.

[8] B. Jaumard and Y. Solari. With the search of shorter second order golomb rulers for cso^{2}c codes. in preparation.

[9] B. Jaumard, Y. Solari, and P. Galinier. On the design of optimum order 2 Golomb ruler. Technical Report G-2003-79, GERAD, Université de Montréal, 2003.

[10] R. Lorentzen and R. Nilsen. Application of linear programming to the optimal difference triangle set problem. *IEEE Transactions on Information Theory*, 37(5):1486–1488, 1991.

[11] C. Meyer and B. Jaumard. Equivalence of some LP-based lower bounds for the *golomb ruler* problem. *Discrete Applied Mathematics*, 154:120–144, 2006.

[12] J. B. Shearer. Improved LP lower bounds for difference triangle sets. *The Electronic Journal of Combinatorics*, 6(R31), 1999.

[13] H. Williams and H. Yan. Representations of the all_different predicate of constraint satisfaction in integer programming. *INFORMS Journal on Computing*, 13(2):96–103, 2001.