Les Cahiers du GERAD ISSN: 0711–2440

On the Design of Optimum Order 2 Golomb Ruler

B. Jaumard, Y. Solari
P. Galinier

G–2003–79

November 2003

On the Design of Optimum Order 2 Golomb Ruler

Brigitte Jaumard
GERAD and Computer Science and Operations Research Department
Université de Montréal and
CRC Chair – Optimization of Communication Networks
jaumard@iro.umontreal.ca

Yannick Solari
Computer Engineering Department
École Polytechnique de Montréal
yannick.solari@polymtl.ca

Philippe Galinier
Computer Engineering Department
École Polytechnique de Montréal
philippe.galinier@polymtl.ca

November, 2003

Les Cahiers du GERAD

G–2003–79

Abstract

A Golomb ruler with M marks can be defined as a set $\{a_i\}$ of integers so that all differences $\delta_{ij} = a_j - a_i, i \neq j$, are distinct. An order 2 Golomb ruler is a Golomb ruler such that all differences $\delta_{ijk\ell} = |\delta_{k\ell} - \delta_{ij}|, \{i, j\} \neq \{k, \ell\}$, are distinct as much as possible. Construction of optimum order 2 Golomb ruler, i.e., of rulers of minimum length, is a highly combinatorial problem which has applications, e.g., in the design of convolutional self doubly orthogonal codes (CSO^{2}C). We propose here a first exact algorithm, different from a pure exhaustive search, for building optimum order 2 Golomb rulers and provide optimal rulers for up to 9 marks and new rulers which improve on the previous ones for up to 20 marks.

Résumé

Une règle de Golomb avec M marques peut être définie comme un ensemble d'entiers $\{a_i\}$ tel que toutes les différences $\delta_{ij} = a_j - a_i, i \neq j$, sont distinctes. Une règle de Golomb d'ordre 2 est une règle de Golomb telle que toutes les différences $\delta_{ijk\ell} = |\delta_{k\ell} - \delta_{ij}|, \{i, j\} \neq \{k, \ell\}$, sont le plus possible distinctes. La construction des règles de Golomb d'ordre 2, c'est-à-dire des règles de longueur minimum, est un problème hautement combinatoire qui a des applications dans, par exemple, la conception de codes doublement orthogonaux convolutionnels (CSO^{2}C). Nous proposons ici un premier algorithme exact, distinct d'une énumération purement exhaustive, pour la construction des règles optimales de Golomb d'ordre 2. Nous construisons également de nouvelles règles qui améliorent les valeurs des règles précédemment connues jusqu'à 20 marques.

1 Introduction

A Golomb ruler with M marks can be defined as a set of M integers $0 \leq a_1 < a_2 < \cdots < a_M$ such that all differences $\delta_{ij} = a_j - a_i$, $1 \leq i < j \leq M$ are distinct. This condition is equivalent to requiring that the sums $a_i + a_j$, $1 \leq i \leq j \leq M$ be distinct. The value a_M defines the length of the ruler. An order 2 Golomb ruler, or 2-Golomb ruler for short, is again defined by a set of M integers $0 \leq a_1 < a_2 < \cdots < a_M$ which satisfy the all difference conditions of Golomb rulers and the additional condition that all differences of differences, or δ-differences for short, $\delta_{ijk\ell} = |\delta_{k\ell} - \delta_{ij}|$, $1 \leq i < j \leq M, 1 \leq k < \ell \leq M, \{i,j\} \neq \{k,\ell\}$ are distinct as much as possible (details are given in Section 2.2). This is in turn equivalent to the condition that, for all i,j,k,ℓ such that $j > \ell \geq i, i \neq k, \ell > k$, differences $\delta_{ij} - \delta_{k\ell} = (a_j - a_i) - (a_\ell - a_k)$ are all distinct. The rulers of interest to researchers are the rulers of minimum length (or of shortest possible length), i.e., such that a_M is minimum for a given number M of marks. Without any loss of generality, we will assume from now on that $a_1 = 0$.

2-Golomb rulers have been recently studied in the context of Turbo decoding using convolutional self doubly orthogonal codes by Cardinal *et al.* [3] and Baechler, Haccoun and Gagnon [2]. It corresponds to a novel coding/decoding technique which circumvents both the complexity and latency shortcomings of the usual Turbo codes. These authors have studied several approaches for designing optimum 2-Golomb rulers, starting from the different approaches that have been developed for Golomb rulers, mainly backtracking algorithms and finite projective geometry methods. Details are provided in Baechler [1]: some 2-Golomb rulers were obtained for M up to 10, but minimum length rulers, with a proof of optimality, were generated only for $M \leq 6$.

We propose to revisit the SHIFT (Dollas *et al.* [4]) and GARSP [5] backtracking algorithms that have been proposed for the design of optimum Golomb rulers. Their success heavily depends on bitmap data structures and their efficient updatings. We generalize them for 2-Golomb rulers. The paper is organized as follows. In Section 2, general properties of 2-Golomb rulers are identified. The data structures and the backtracking algorithm are described in Section 3. Computational experiences are discussed in Section 4, and conclusions are drawn in the last section.

2 Unavoidable Repetitions and Properties of 2-Golomb Rulers

2.1 Definitions and Notation

Let Δ be the set of differences of a Golomb ruler. It can be represented in a difference triangle, denoted by Δ-triangle, as illustrated on a 5-mark Golomb ruler on Figure 1. The first two lines correspond to the mark positions on the ruler, the third line to the differences between two consecutive marks, i.e. 0-hop differences, the fourth line to 1-hop differences, i.e., differences of the type $a_{i+2} - a_i$, and so on.

a_1		a_2		a_3		a_4		a_5
0		**11**		**12**		**54**		**57**
	11		1		42		3	
		12		43		45		
			54		46			
				57				

Figure 1: The Δ-triangle representation of a 5-mark Golomb ruler

We have generalized the difference triangle for 2-Golomb rulers. Let $\Delta\Delta$ be the set of differences of differences of a 2-Golomb ruler, also called set of δ-differences. Again, it can be represented in a difference of difference triangle, denoted by $\Delta\Delta$-triangle, where the first two lines correspond to the ordered set of differences δ_{12}, δ_{23}, δ_{13}, δ_{34}, δ_{24}, $\delta_{14}, \ldots, \delta_{M-1,M}$, $\delta_{M-2,M}, \ldots, \delta_{1,M}$ with $\delta_{ij} = a_j - a_i$, the third line corresponds to differences of differences with a 0-hop, i.e., of two consecutive differences, the fourth line corresponds to differences of differences with a 1-hop, and so on. The $\Delta\Delta$-triangle of a 2-Golomb ruler with 5 marks associated with the Δ-triangle of Figure 1, is represented on Figure 2.

δ_{12}		δ_{23}		δ_{13}		δ_{34}		δ_{24}		δ_{14}		δ_{45}		δ_{35}		δ_{25}		δ_{15}
11		**1**		**12**		**42**		**43**		**54**		**3**		**45**		**46**		**57**
	10		$\underline{11}$		30		$\underline{1}$		$\underline{11}$		51		$\underline{42}$		$\underline{1}$		$\underline{11}$	
		$\underline{1}$		41		$\underline{31}$		$\underline{12}$		40		$\underline{9}$		$\underline{43}$		$\underline{12}$		
			$\underline{31}$		$\underline{42}$		$\underline{42}$		39		$\underline{2}$		$\underline{8}$		54			
				32		53		$\underline{9}$		$\underline{3}$		$\underline{3}$		$\underline{3}$				
					$\underline{43}$		$\underline{2}$		33		4		14					
						$\underline{8}$		44		$\underline{34}$		15						
							$\underline{34}$		$\underline{45}$		$\underline{45}$							
								35		36								
									46									

Figure 2: The $\Delta\Delta$-triangle representation of a 5-mark 2-Golomb ruler (repeated δ-differences are underlined)

2.2 Unavoidable Repetitions

Unlike the Δ-triangle, the $\Delta\Delta$-triangle contains some unavoidable repetitions. This point was outlined in Baechler [1], but not thoroughly addressed. Indeed, there are two types of repetitions that we characterize and take a census of below. For a given pair $i, k \in \{1, 2, \ldots, M\}$ of distinct indices, a first set of repetitions, the repetitions of type 1, corresponds to the set of differences

$$\begin{aligned}
\delta_{ij} - \delta_{kj} &= a_j - a_i - (a_j - a_k) \\
&= a_k - a_i, \qquad j = 2, 3, \ldots, M;\ i < j;\ k < j, \\
\delta_{ji} - \delta_{jk} &= a_i - a_j - (a_k - a_j) \\
&= a_i - a_k, \qquad j = 1, 2, \ldots, M-1;\ j < i;\ j < k.
\end{aligned}$$

We say that such a repetition is of order t if the value $|a_k - a_i|$ cannot be encountered less than t times in a given $\Delta\Delta$-triangle. It is easy to verify that there are at least $M-t-1$ differences of order $M-t$ for $1 \leq t \leq M-2$. There is a second type of repetitions which corresponds to the following ones. Given a 4-uple (i,j,k,ℓ), $1 \leq i < j < k < \ell \leq M$, consider the set of δ-differences:

$$\begin{aligned}\delta_{k\ell} - \delta_{ij} &= (a_\ell - a_k) - (a_j - a_i)\\ &= a_\ell - a_k - a_j + a_i\\ \delta_{j\ell} - \delta_{ik} &= (a_\ell - a_j) - (a_k - a_i)\\ &= a_\ell - a_k - a_j + a_i.\end{aligned}$$

There are $\binom{M}{4}$ such repetitions.

We summarize in Table 1 the number of unavoidable repetitions for 2-Golomb rulers ($3 \leq M \leq 10$). We denote by $N_{\Delta\Delta}$ the overall number of δ-differences and by $N^d_{\Delta\Delta}$ the number of distinct δ-differences. $N_{\Delta\Delta} = \frac{N_\Delta(N_\Delta - 1)}{2}$ where $N_\Delta = \frac{M(M-1)}{2}$ is the number of differences, i.e., of elements in a Δ-triangle. We then have:

$$N^d_{\Delta\Delta} = N_{\Delta\Delta} - \sum_{t=1}^{M-2}(M-t)(M-t-2) - \binom{M}{4}.$$

M	t							N_Δ	$N_{\Delta\Delta}$	$N^d_{\Delta\Delta}$
	2	3	4	5	6	7	8			
3	-	-	-	-	-	-	-	3	3	3
4	3+ 1	-	-	-	-	-	-	6	15	11
5	3+ 5	4	-	-	-	-	-	10	45	29
6	3+ 15	4	5	-	-	-	-	15	105	64
7	3+ 35	4	5	6	-	-	-	21	210	125
8	3+ 70	4	5	6	7	-	-	28	378	223
9	3+ 126	4	5	6	7	8	-	36	630	371
10	3+ 210	4	5	6	7	8	9	45	990	584

Table 1: Counting the number of unavoidable repetitions

The comparison of the values of N_Δ and $N^d_{\Delta\Delta}$ leads to expect a huge increase in the combinatorics of finding optimal 2-Golomb rulers that the computational results will confirm in practice, see Section 4.

2.3 Structure of the $\Delta\Delta$-triangle

Any sub-ruler of a 2-Golomb ruler is a 2-Golomb ruler. Let us consider the sub-ruler made of the first m marks. The associated $\Delta\Delta$-triangle can be partitioned into three sub-triangles as follows (see Figure 3 for an illustration):

- $\Delta\Delta^L$-triangle or left sub-triangle, denoted $\Delta\Delta^L_m$: the δ-differences are of the form $|(a_j - a_i) - (a_k - a_\ell)|$, $i < j < m, \ell < k < m$.
- $\Delta\Delta^C$-triangle or center sub-triangle, denoted $\Delta\Delta^C_m$: the δ-differences are of the form $|(a_m - a_i) - (a_\ell - a_k)|$, $i < m, k < \ell < m$;
- $\Delta\Delta^R$-triangle or right sub-triangle, denoted $\Delta\Delta^R_m$: the δ-differences are of the form $|(a_m - a_i) - (a_m - a_\ell)| = |a_\ell - a_i|$, $i \neq \ell, i < m, \ell < m$.

We have the following properties.

Proposition 1 *All elements of the $\Delta\Delta^C$-triangle must be different from the elements of the $\Delta\Delta^L$ and $\Delta\Delta^R$ triangles in order to define a 2-Golomb ruler.*

Proof. Elements of the $\Delta\Delta^C$-triangle are the only elements that depend on a_m and can all be written under the form $a_m \pm a$ with $a > 0$. Therefore taking a_m large enough makes it always possible to avoid repetitions with elements of $\Delta\Delta^L \cup \Delta\Delta^R$. □

Proposition 2 *All elements of the $\Delta\Delta^R$ triangle belong to the $\Delta\Delta^L$ triangle except for one element.*

Proof. First observe that the elements of $\Delta\Delta^L$ contains all the differences except for $\delta_{m-1,1}$. Consider the δ-differences of $\Delta\Delta^L$: $|(a_j - a_i) - (a_k - a_\ell)|$, $i < j < m, \ell < k < m$. When $j = k$, we can obtain all differences

$$\delta_{i\ell} = a_\ell - a_i \text{ for } i \neq \ell < m - 1. \tag{1}$$

When $i = \ell$, we can obtain all differences $a_j - a_k$ for $1 < j \neq k < m$, therefore in particular all differences

$$\delta_{k,m-1} = a_{m-1} - a_k \text{ for } k = 2, 3, \ldots, m - 2. \tag{2}$$

Now considering the δ-differences of $\Delta\Delta^R$, they are indeed differences including $\delta_{1,m-1}$, hence the result. □

2.4 Symmetry

The ruler $\mathcal{R}' = [a'_1, a'_2, \ldots, a'_M]$ is a symmetrical ruler of the ruler $\mathcal{R} = [a_1, a_2, \ldots, a_M]$ if $a'_i = a_M - a_{M-i+1}$ for all $i = 1, 2, \ldots, M$. For instance $\mathcal{R} = [0 \quad 1\ 9\ 12]$ and $\mathcal{R}' = [0\ 3\ 11\ 12]$ are symmetric rulers. In order to be efficient when building a ruler, it is then wise to eliminate the generation of symmetrical rulers. Depending of the search scheme that is used, there are different ways to avoid generating symmetrical rulers. We describe some of them below assuming we consider a depth first search scheme in which we fix a new mark at each step, going from left to right on the ruler. Consider a 2-Golomb ruler with M marks.

Test 1 *Impose that $d_{12} > d_{M-1,M}$.*

Test 2 *Impose that $d_{M-1,M} > d_{12}$.*

Assuming we consider a search scheme in which we first determine a value for d_{12} and in the last iteration the value of $d_{M-1,M}$, these last two tests have the drawback that we must wait until the last iteration before starting to eliminate the generation of symmetrical rulers.

Test 3 *Impose that the central difference of two consecutive marks if M is even, or the second central difference of two consecutive differences if M is odd, be smaller than its previous consecutive difference between two consecutive marks:* $d_{\lfloor M/2+1/2\rfloor,\lfloor M/2+1/2\rfloor-1} > d_{\lfloor M/2+1\rfloor+1,\lfloor M/2+1\rfloor}$.

Test 4 *This test is similar to the previous one, except that the direction of the inequality is reversed:* $d_{\lfloor M/2+1\rfloor+1,\lfloor M/2+1\rfloor} > d_{\lfloor M/2+1/2\rfloor,\lfloor M/2+1/2\rfloor-1}$.

Test 5 *Impose that the first half of the ruler be not smaller than the second half:* $d_{\lfloor M/2+1/2\rfloor,1} > d_{M,\lfloor M/2+1\rfloor}$.

Test 6 *Impose that the first half of the ruler be not larger than the second half:* $d_{\lfloor M/2+1/2\rfloor,1} < d_{M,\lfloor M/2+1\rfloor}$.

We compare the six symmetry tests on the search of an optimal 2-Golomb ruler with 6 marks and the results are presented in Table 2. For each solution with a given symmetry test, we have indicated the overall number of nodes and leafs in the search tree in a simple depth first search scheme, as well as computing times (Pentium II, 550 MHz). We observe that tests 2, 5 and 6 are the most interesting ones as they reduce the size of the search tree by more than 50 % with respect to the other symmetry tests. In the sequel, we will use the symmetry test 6.

	no test	test 1	test 2	test 3	test 4	test 5	test 6
# nodes	524 937	373 548	222 363	327 321	293 713	219 745	193 315
# leafs	214 420	140 844	84 678	115 612	98 808	86 459	84 321
cpu (sec.)	112	80	47	71	60	48	41

Table 2: Comparison of symmetry tests

3 A First Algorithm

3.1 Data Structures

We started from the same data structures than those used in the GARSP algorithm. They are described in the paper of Dollas *et al.* [4] for an early version, called SHIFT, of the GARSP algorithm. The SHIFT algorithm makes use of three bitmap structures: LIST for the new differences generated after the addition of a new mark, DIST for storing all measured differences, and COMP for computing the first available position for the next mark. We generalize those structures for the informations related to the differences of differences.

The resulting O2GOLRU algorithm for 2-Golomb rulers is described in Section 3.3. It corresponds to an iterative depth first search algorithm in which at each iteration, a new mark denoted a_m is defined, assuming we fix the marks, going from left to right on the ruler. In order to simplify the presentation, we first describe the data structures using a set formalism, and then explain how to convert them in a bitmap formalism for an efficient implementation of the O2GOLRU algorithm. We also discuss the updatings of the data structures at iteration m, when mark m is inserted in position a_m on the ruler.

Given sets S, S_1 and S_2, we will use the following notation:

$$\begin{aligned} S_1 \oplus S_2 &= \{s_1 + s_2, s_1 \in S_1, s_2 \in S_2\}, \\ S_1 \ominus S_2 &= \{s_1 - s_2, s_1 \in S_1, s_2 \in S_2\}, \\ |S| &= \{|s|, s \in S\}. \end{aligned}$$

3.1.1 The LIST structures. At each iteration, we store the set of new differences generated by the addition of mark m at position a_m in the ruler in the structure LIST_m. The updating of the structure is as follows:

$$\text{LIST}_m \leftarrow (\text{LIST}_{m-1} \oplus \{\delta_{m-1,m}\}) \cup \{\delta_{m-1,m}\}. \tag{3}$$

Considering the example with 5 marks described in Figure 1, we have $\text{LIST}_3 = \{1, 12\}$, $\text{LIST}_4 = \{42, 43, 54\}$ and $\text{LIST}_5 = \{3, 45, 46, 57\}$.

3.1.2 The DIST structures. The list of differences is also stored in a list structure, denoted by DIST_m. Each element in DIST_m represents a distance between two marks, measured by the current ruler. Updating of DIST_m goes as follows:

$$\text{DIST}_m \leftarrow \text{DIST}_{m-1} \cup \text{LIST}_m.$$

Going on with the same example, we have $\text{DIST}_3 = \{1, 11, 12\}$, $\text{DIST}_4 = \{1, 11, 12, 42, 43, 54\}$ and $\text{DIST}_5 = \{1, 3, 11, 12, 42, 43, 45, 46, 54, 57\}$.

3.1.3 The DLIST structures. For the differences of differences, we introduce a first structure DLIST_m to store the new differences of differences that are generated after the addition of mark a_m. In order to simplify the notation, let us denote $\delta_{m-1,m}$ by δ throughout this section.

Proposition 3 *Updating of the* DLIST_m *structure.*

$$\text{DLIST}_m \leftarrow \Big(\text{DLIST}_{m-1} \oplus \{\delta\}\Big) \cup \Big(\{\delta\} \ominus \text{DIST}_{m-1}\Big) \cup \{\delta, a_{m-1}\}.$$

Proof. Let us study the structure of the $\Delta\Delta_m$-triangle and let us partition it as illustrated in Figure 3.

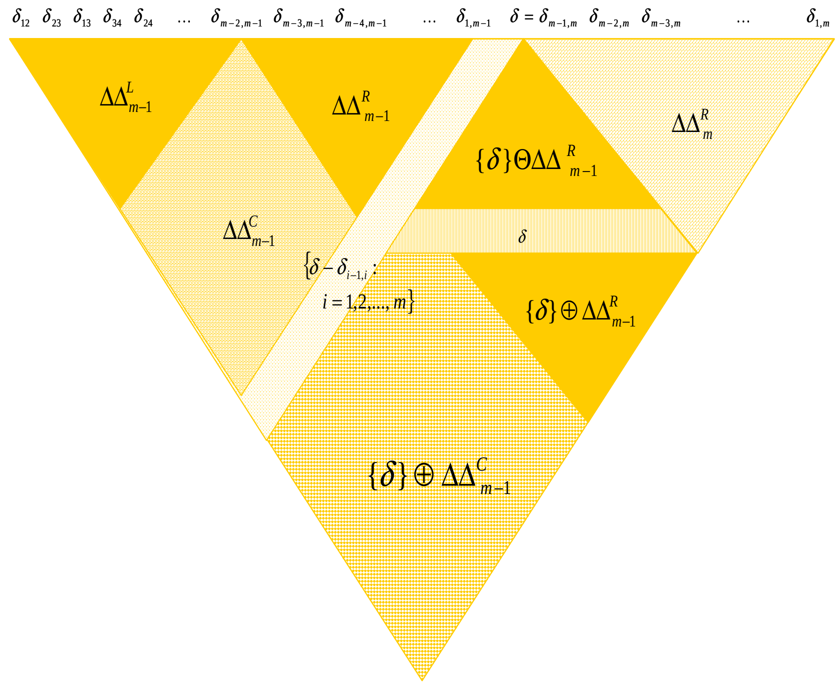

Figure 3: Partition of the $\Delta\Delta_m$-triangle

Observe first that DLIST_m is indeed made of the elements of $\Delta\Delta^C_m \cup \Delta\Delta^R_m = \Delta\Delta^C_m \cup \text{DIST}_{m-1}$. However, $\text{DIST}_{m-1} \subseteq \Delta\Delta^L_{m-1} \cup \{\delta_{1,m-1}\}$ (due to Proposition 2). Note that $\delta_{1,m-1} = a_{m-1}$.

Let $\delta_{im} - \delta_{k\ell} = (a_m - a_i) - (a_\ell - a_k)$ be an element of $\Delta\Delta^C_m$, i.e., such that $i < m$ and $k < \ell < m$. We can rewrite it:

$$\delta_{im} - \delta_{k\ell} = (a_m - a_i) - (a_\ell - a_k) = (a_m - a_{m-1}) + (a_{m-1} - a_i) - (a_\ell - a_k).$$

We can partition $\Delta\Delta^C_m$ into five sets:
$\Delta\Delta^{C1}_m = \{\delta_{im} - \delta_{k\ell} \in \Delta\Delta^C_m : i = m-1\}$,
$\Delta\Delta^{C2}_m = \{\delta_{im} - \delta_{k\ell} \in \Delta\Delta^C_m : i < m-1; \ell = m-1; k = i\}$,
$\Delta\Delta^{C3}_m = \{\delta_{im} - \delta_{k\ell} \in \Delta\Delta^C_m : i < m-1; \ell = m-1; i < k\}$,
$\Delta\Delta^{C4}_m = \{\delta_{im} - \delta_{k\ell} \in \Delta\Delta^C_m : i < m-1; \ell = m-1; i > k\}$,
$\Delta\Delta^{C5}_m = \{\delta_{im} - \delta_{k\ell} \in \Delta\Delta^C_m : i < m-1; \ell < m-1\}$.
Studying carefully the elements of those sets, we can see that:
$\Delta\Delta^{C1}_m = \{\delta\} \ominus \text{DIST}_{m-1}$,
$\Delta\Delta^{C2}_m = \{\delta\}$,
$\Delta\Delta^{C3}_m = \{\delta\} \ominus \Delta\Delta^R_{m-1} = \{\delta\} \ominus \text{DIST}_{m-2} \subseteq \{\delta\} \ominus \text{DIST}_{m-1}$,
$\Delta\Delta^{C4}_m = \{\delta\} \oplus \Delta\Delta^R_{m-1} = \{\delta\} \oplus \text{DIST}_{m-2}$,
$\Delta\Delta^{C5}_m = \{\delta\} \oplus \Delta\Delta^C_{m-1}$.

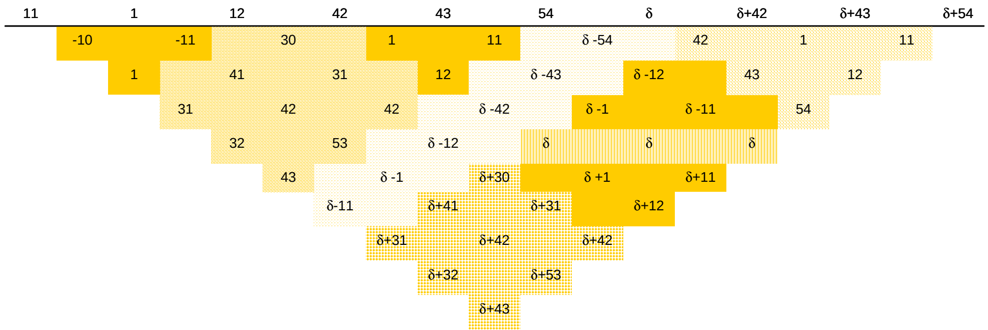

Figure 4: Illustration of the $\Delta\Delta_m$-triangle partition of a 2-Golomb ruler of 5 marks

Therefore $\Delta\Delta_m^{C4} \cup \Delta\Delta_m^{C5} = \{\delta\} \oplus \left(\textsc{dist}_{m-2} \cup \Delta\Delta_{m-1}^{C}\right) = \{\delta\} \oplus \textsc{dlist}_{m-1}$. This concludes the proof. □

Let us now introduce the $\textsc{dnext}_m$ structure that will be used later in some proof:

$$\textsc{dnext}_m = \Big(\textsc{dlist}_m \cup \big(\{0\} \ominus \textsc{dist}_m\big) \cup \{0\}\Big), \tag{4}$$

leading to:

$$\textsc{dlist}_m = \Big(\textsc{dnext}_{m-1} \oplus \{\delta\}\Big) \cup \{a_{m-1}\}. \tag{5}$$

3.1.4 The DDIST structures. We also define the list structure $\textsc{ddist}_m$ to store all differences of differences already present in the ruler, and its updatings is as follows:

$$\textsc{ddist}_m \leftarrow \textsc{ddist}_{m-1} \cup \textsc{dlist}_m.$$

Before discussing the last structures, let us re-express the structure of the $\Delta\Delta$-triangle in terms of the structures we have introduced so far. Using the definitions of the structures $\textsc{dist}_m$, $\textsc{list}_m$, $\textsc{ddist}_m$ and $\textsc{dlist}_m$, it is easy to see that the elements of the sub-triangles $\Delta\Delta_m^L$, $\Delta\Delta_m^C$ and $\Delta\Delta_m^R$ correspond to the sets $\textsc{ddist}_{m-1}$, $\textsc{list}_m \ominus \textsc{dist}_{m-1}$ and $\textsc{dist}_{m-1}$ respectively, as illustrated on Figure 5 below.

3.1.5 The DCOMP structures. At the m^{th} iteration, the $\textsc{dcomp}_m$ structure contains the positions which are forbidden for the next mark to be defined, so that the first available position a_{m+1} for the $(m+1)^{\text{th}}$ mark corresponds to a feasible 2-Golomb ruler. DCOMP contains the forbidden positions due to repetitions of δ-differences, and takes care of unavoidable repetitions of differences of differences. The updating of DCOMP is quite complex and can be decomposed in 3 steps that are described in the next theorem.

Theorem 1 *Updating of the* $\textsc{dcomp}_m$ *structure can be obtained by performing the following three steps:*

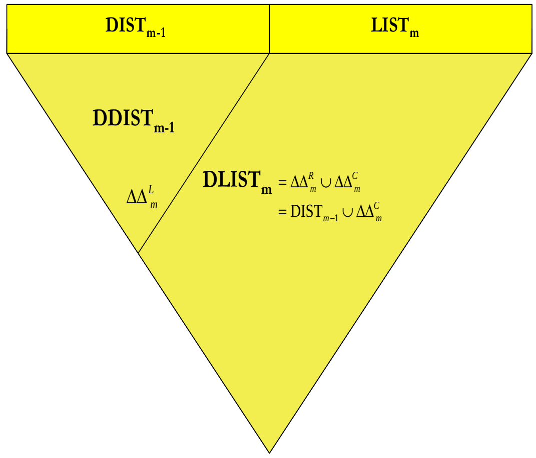

Figure 5: Another view of the partition of the $\Delta\Delta_m$-triangle

$$\begin{aligned}
&\textit{Step 1}: \textsc{dcomp}_m \leftarrow \textsc{dist}_m \oplus \Big(\textsc{ddist}_m \cup -\textsc{ddist}_m \\
&\qquad\qquad \cup\{-\delta_{1,m}, 0, \delta_{1,m}\}\Big) \\
&\textit{Step 2}: \textsc{dcomp}_m \leftarrow \textsc{dcomp}_m \ominus (\textsc{list}_m \cup \{0\}) \\
&\textit{Step 3}: \textsc{dcomp}_m \leftarrow \textsc{dcomp}_m \cup \tfrac{1}{2}\Big(\textsc{dnext}_m \oplus \textsc{dnext}_m\Big).\ ^{1}
\end{aligned}$$

Proof. Observe first that assuming $\Delta\Delta_m$ to be known, all values of $\Delta\Delta_{m+1}$ are either constant values that are independent of a_{m+1}, i.e., the elements of $\Delta\Delta^R_{m+1}$ and $\Delta\Delta^L_{m+1}$, or values of the form $\delta_{m-1,m} \pm a$ where a is independent of a_{m+1}, i.e., the elements of $\Delta\Delta^C_{m+1}$. Therefore the updating of $\textsc{dcomp}_m$ must lead to a $\Delta\Delta_{m+1}$-triangle with the minimum number of unavoidable repetitions among the values of $\Delta\Delta^C_{m+1}$.

We now justify each of the steps of the updating.

Step 1. Let us first verify the condition of Proposition 1: we must guarantee that, when determining a value for a_{m+1}, we will have $\Delta\Delta^C_{m+1} \cap (\Delta\Delta^R_{m+1} \cup \Delta\Delta^L_{m+1}) = \emptyset$ or equivalently that $\Big(\textsc{list}_{m+1} \ominus \textsc{dist}_m\Big) \cap \Big(\textsc{ddist}_m \cup \textsc{dist}_m\Big) = \emptyset$. Let $\delta\delta$ be an element of $\Delta\Delta^C_{m+1}$: it can be written

$$\delta\delta = |\delta_{i,m+1} - \delta_{k\ell}|, \qquad k < \ell < m+1; i = 1, 2, \ldots, m$$

where $|\delta_{k\ell}| \in \textsc{dist}_m$. We must therefore forbid

$$|\delta_{i,m+1} - \delta_{k\ell}| \in \textsc{ddist}_m \cup \textsc{dist}_m,$$

or equivalently

$$\delta_{i,m+1} - \delta_{k\ell} \notin \textsc{ddist}_m \cup -\textsc{ddist}_m \cup \textsc{dist}_m \cup -\textsc{dist}_m,$$

[1] We assume here that $\frac{1}{2}(S_1 \oplus S_2) = \{s = (s_1 + s_2)/2, s \text{ is a nonnegative integer}, s_1 \in S_1, s_2 \in S_2\}$.

or again

$$\delta_{i,m+1} \notin \text{DIST}_m \oplus \Big(\text{DDIST}_m \cup -\text{DDIST}_m \cup \{\delta_{1,m}, -\delta_{1,m}\}\Big)$$

as $\text{DIST}_m \subseteq \text{DDIST}_m \cup \{\delta_{1,m}\}$. Moreover, $\delta_{i,m+1}$ cannot take its value in DIST_m, hence the result:

$$\delta_{i,m+1} \notin \text{DIST}_m \oplus \Big(\text{DDIST}_m \cup -\text{DDIST}_m \cup \{\delta_{1,m}, 0, -\delta_{1,m}\}\Big).$$

In other words, DCOMP_m now contains all values that are forbidden for LIST_{m+1} in order that $\Delta\Delta^C_{m+1} \cap (\Delta\Delta^R_{m+1} \cup \Delta\Delta^L_{m+1}) = \emptyset$.

Step 2. Considering the updating formula (3) of LIST_{m+1}, one observes that once δ is fixed, all values of LIST_{m+1} are also fixed. Therefore, one must be cautious about the fact that even if a value $\delta + a$ with $a \in \text{LIST}_m$ is a feasible value (i.e., does not belong to DCOMP_m), it does not necessarily entail that this is the case for all values $\delta + a', a' \neq a, a' \in \text{LIST}_m$. In order to guarantee that this is indeed the case, we must perform the following operation:

$$\text{DCOMP} \leftarrow \text{DCOMP} \ominus \big(\text{LIST}_m \cup \{0\}\big).$$

Step 3. Let us first examine how to avoid repetitions between absolute values of $\Delta\Delta^{C+}_m$, i.e. the elements of $\Delta\Delta^C_m$ of the form $\delta + \delta'$, $\delta' > 0$ (i.e. $\Delta\Delta^{C4}_m \cup \Delta\Delta^{C5}_m$), and the values of $\Delta\Delta^{C-}_m$, i.e. the elements of $\Delta\Delta^C_m$ of the form $\delta + \delta''$, $\delta'' < 0$ (i.e. $\Delta\Delta^{C1}_m \cup \Delta\Delta^{C3}_m$). Observe first that we can have two absolute values of $\Delta\Delta^{C-}_m$ that are equal: take δ, δ''_1, δ''_2 such that $\delta + \delta''_1 = -(\delta + \delta''_2)$ (e.g., $\delta''_1 = -1$, $\delta''_2 = -3$ and $\delta = 2$: we have 2 - 1 = - (2 - 3)). We can also have an absolute value of $\Delta\Delta^{C+}_m$ equal to an absolute value of $\Delta\Delta^{C-}_m$: take δ, δ' and δ'' such that $\delta + \delta'' = -(\delta + \delta')$, i.e., $2\delta = -(\delta'' + \delta')$.

These repetitions are avoided by removing (i.e. adding to DCOMP) the nonnegative integer values of

$$\frac{1}{2}\Big(\text{DNEXT}_m \oplus \text{DNEXT}_m\Big),$$

where $\text{DNEXT}_m = \Big(\text{DLIST}_m \cup \big(\{0\} \ominus \text{DIST}_m\big) \cup \{0\}\Big)$, see (4). Observing that $\Delta\Delta^C_m = \text{DLIST}_m \setminus a_{m-1}$ (see, e.g., the proof of Proposition 3) and the expression (5) of DLIST_m, it is easy to verify that $\Delta\Delta^C_m = \text{DNEXT}_m + \{\delta\}$ which completes the proof. □

3.2 Lower Bounds

Lower bounds are extremely useful in order to fathom nodes in the search tree of an implicit enumeration scheme, i.e., when the lower bound of the current subproblem is larger than or equal to the incumbent solution (i.e, the best known solution). However, accurate lower bounds are difficult to compute, and as in the construction of Golomb rulers, too costly to compute in order to efficiently fathom nodes (see, e.g., Hoa [6]). We have considered the following 5 lower bound options. Let a^{best}_M be an upper bound on the optimum length.

LB0. No lower bounds.

LB1. Use the previous optimum values for 2-Golomb rulers: we must have $a_m \leq a_M^{\text{best}} - a_\ell^*$ where a_ℓ^* is the (proved) optimum 2-Golomb value with ℓ marks, $\ell = M - m$. As optimal 2-Golomb rulers are only known for $\ell \leq 9$, we can use this test only for $m \geq M - 9$.

LB2. Build a database which considers three marks, their set of differences and δ-differences. Consider that you want to add a new mark and determines the first available position for this new mark, the second and third next following ones. Coming back to the search of an optimum M 2-Golomb ruler, if a_m is the current mark, use the values of this database in order to compute a lower bound for $a_{m+1}, a_{m+2}, a_{m+3}$ and possibly curtail the search. A lot of computational effort (several days of computing) was required to compute this database.

LB3. Considering the huge computational effort that would be required to enrich the database used in LB2, we considered the weaker, but available, lower bounds of the database built and used in the GARSP algorithm [5] for the design of optimum Golomb rulers. The database consists in the optimum values of Golomb rulers, the length M of the ruler being given, as well as a set of forbidden distance values, for $M \leq 11$.

LB4. No use of precomputed database, but an on-line computation of the minimum length of the right sub-ruler $[a_{m+1}, a_{m+2}, \ldots, a_M]$, say $\tilde{a}_{M-m}$, assuming the differences and δ-differences already used in the left part of the ruler are forbidden. We must have $a_m \leq a_M^{\text{best}} - \tilde{a}_{M-m}$.

3.3 Algorithm

We describe in this section the algorithm used to search for optimum 2-Golomb rulers. It is a depth first search procedure which attempts to find mark positions, going from the left to the right of the ruler. If at some step, the current length of the ruler is larger than the lower bound, a backtrack is performed.

Algorithm O2GOLRU

Initialization

$a_1 \leftarrow 0; a_2 \leftarrow 1;$ $m \leftarrow 3$;
Set the structures LIST_m, DIST_{m-1}, DLIST_m, DDIST_{m-1} and DCOMP_m;
Select the lower bound computation scheme: let LB_m be the current lower bound;
Select the symmetry test ;
Compute a first upper bound on the optimum length a_M^{best} (or set a_M^{best} to $+\infty$);

Current iteration

While $m > 0$ do
  $a_m \leftarrow$ first position available for an additional mark,
    deduced from DCOMP_{m-1};
  If symmetry test is true then perform a backtracking ;
  If $a_m \geq a_M^{\text{best}} - \text{LB}_m$ then perform a backtracking ;
  If $m = M - 1$ then
    if we have a 2-Golomb ruler then

update, if necessary, the incumbent 2-Golomb ruler ;
perform a backtracking
Update the sets LIST_m, DIST_m, DLIST_m, DDIST_m
and DCOMP_m ;
$m \leftarrow m + 1$;
Compute the lower bound LB_m ;

In order to improve the speed of the O2GOLRU algorithm, the various sets LIST_m, DIST_m, DLIST_m, DDIST_m and DCOMP_m have been implemented using bitmap structures. Then the updating operations can be performed much faster and we illustrate it with the LIST_m structure. It is then easy to generalize it to the other structures converting the $\oplus$ to a shift on the right, the $\ominus$ to a shift on the left, and the $\cup$ and $\cap$ set operations with the $\vee$ and $\wedge$ logical operations.

Let us consider the updating of the LIST structure with a bitmap fashion: the i^{th} bit is equal to 1 if i measures a new difference. The updating of LIST_m can then be restated as follows in terms of bitmaps:

$$\text{LIST}_m \leftarrow \delta \vee (\text{LIST}_{m-1} >> \delta),$$

assuming δ is also represented using a bitmap structure with the δ^{th} bit equal to 1, and all the other bits equal to 0.

4 Results

The algorithm O2GOLRU has been implemented in C++ using the bitmap operations as described in the previous section. In addition, several bitmap operations were specifically designed for speeding up the updating of the DCOMP structure, and their details can be found in Solari [8]. We first report in Table 3 the best order 2 Golomb rulers that we were able to find, running the program on a Pentium II, 550MHz. For $M \leq 9$, the rulers are optimum as the algorithm O2GOLRU stops after a reasonable amount of time. For $M = 9$ proving the optimality required using 95 workstations (45 1GHz, 50 500 MHz, i.e., roughly 3.63 years on a single machine) and decomposing the search so as to simulate a parallel implementation. For $10 \leq M \leq 12$, we were able to improve significantly the best previous results, due to Baechler [1] (Table 4.13 p. 97 in his M.Sc. thesis) and Baechler, Haccoun and Gagnon [2] for $M = 10$ and 11, and to Cardinal *et al.* [3] for $M = 9$ and 12. At least for $M = 6$ and 8, there are two optimum order 2 Golomb rulers which are described in Table 3. However, we have omitted the symmetrical rulers that can be deduced from each optimum ruler, e.g., for $M = 4$ the symmetrical ruler of $[0, 1, 9, 12]$ is $[0, 3, 11, 12]$. Although for $M \geq 10$, the listed rulers are probably not optimum, we have compared in Figure 6 the length growth of the Golomb and 2-Golomb rulers. For Golomb rulers, it is well known that the length growth behaves similarly to M^2 (curve labeled $M * M$ on the graph of Figure 6). While for $M \leq 9$, length growth is reasonable and similar to M^3, it is

# marks	Previous results	Algorithm O2GOLRU	Optimum and best known rulers obtained with O2GOLRU
3	4⋆ [1]	4⋆	0, 1, 4
4	12⋆ [1]	12⋆	0, 1, 9, 12
5	33⋆ [1]	33⋆	0, 1, 4, 24, 33
6	79⋆ [1]	79⋆	0, 2, 53, 58, 73, 79 0, 2, 53, 59, 74, 79
7	174⋆ [1]	174⋆	0, 4, 42, 55, 156, 171, 174
8	375 [1]	345⋆	0, 3, 30, 39, 238, 264, 334, 345 0, 3, 30, 39, 248, 264, 344, 345
9	913 [1], 863 [2]	666⋆	0, 1, 37, 76, 91, 464, 492, 661, 666
10	1 698 [1]	1 329	0, 1, 4, 13, 40, 96, 368, 920, 1115, 1329
11	5 173 [1]	2 315	0, 1, 4, 13, 40, 96, 368, 1372, 1906, 2120, 2315
12	4 951 [2]	4 254	0, 7, 18, 87, 88, 130, 282, 902, 1422, 3231, 3754, 4254
13	-	6 997	0, 1, 4, 13, 40, 96, 213, 471, 1229, 3029, 4891, 6388, 6997
14	-	11 372	0, 1, 4, 13, 40, 96, 213, 471, 1004, 2276, 5138, 8399, 9960, 11372
15	-	18 975	0, 3, 81, 82, 90, 112, 287, 613, 1284, 2167, 3992, 8442, 14052, 17523, 18975
16	-	27 057	0, 3, 4, 13, 37, 109, 258, 558, 923, 1900, 3918, 6307, 13001, 17506, 24840, 27057
17	-	42 133	0, 2, 5, 14, 41, 103, 246, 536, 1140, 2388, 3833, 5552, 9641, 17986, 30510, 37338, 42133
18	-	61 964	0, 2, 5, 14, 41, 103, 246, 536, 1140, 2388, 3833, 5552, 9641, 15815, 34224, 44404, 55676, 61964
19	-	88 652	0, 4, 5, 16, 41, 100, 245, 552, 1181, 1975, 3570, 6123, 10397, 14818, 21978, 47833, 58968, 71518, 88652
20	-	124 588	0, 4, 5, 16, 41, 100, 245, 552, 1181, 1975, 3570, 6123, 10397, 14818, 21978, 34528, 61233, 79091, 107169, 124588

Table 3: Comparison of Baechler's and O2GOLRU's 2-Golomb Rulers

much faster for larger value of M. Note however that the best optimum lengths may be much smaller than the best known today.

We have also performed some tests to evaluate the usefulness of the lower bounds. It is well known for the search of Golomb rulers that easy to compute lower bounds are helping exact algorithms. Moreover, even if we forget the computational effort, sharp lower bounds are difficult to compute. This is even more so for 2-Golomb rulers. We compare the 5 lower bounds options described in Section 3.2 on the search of 3 rulers: the optimum ruler for $M = 7$, the best ruler assuming some preselected values for the first three marks for $M = 8$, and for the first 7 marks for $M = 11$. For each ruler, we provide the number of nodes at given levels of the search tree as well as the overall number of leafs. We use the lower bound LB1 as a reference for computing the percentages. As expected, the LB2 bound is the most promising for small rulers with an efficiency that decreases drastically when the number of marks is increasing. The LB3 bounds, even if very efficient for Golomb rulers, are not very useful because they are too weak. Finally, the lower bound LB4 is probably the most interesting one, even if not the most efficient one here for a relatively small number of marks. It provides a lower bound computing scheme that can be easily used for any mark without a significant change in the computational effort, and still a significant impact on the efficiency in the pruning of the search tree.

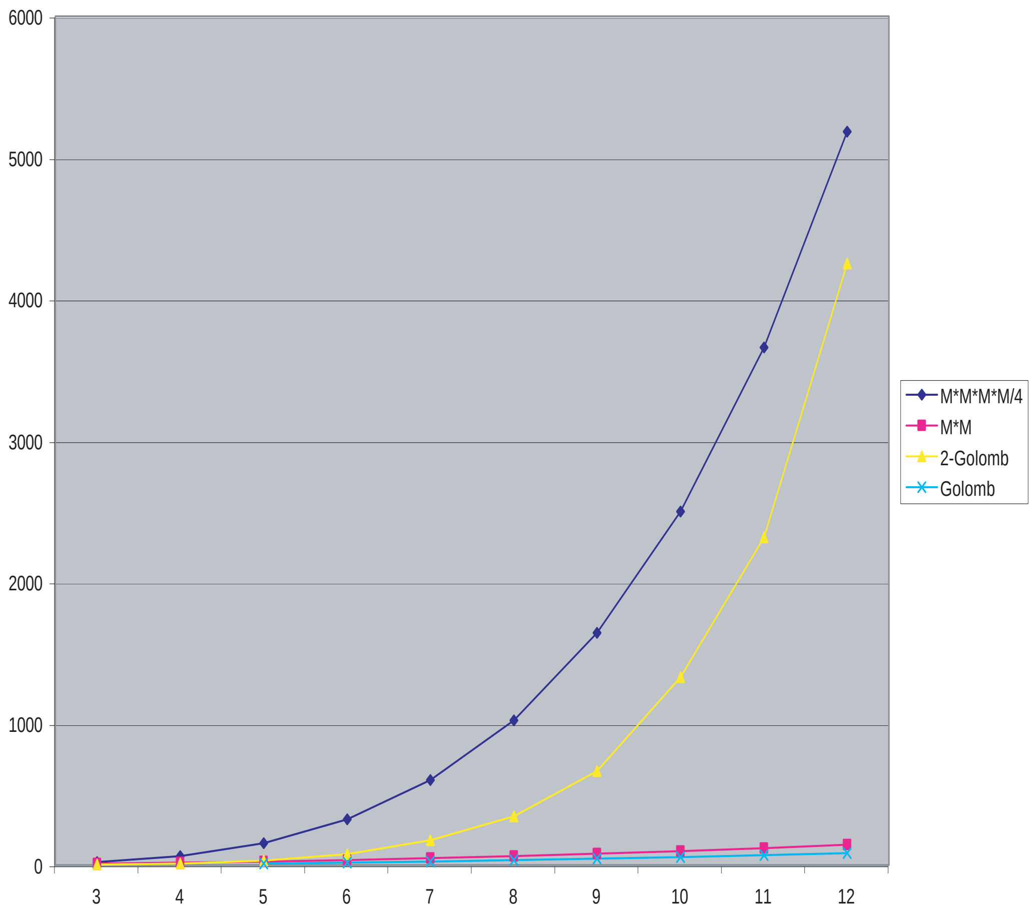

Figure 6: Length Growth of the Golomb and 2-Golomb rulers

2-Golomb ruler with $M = 7$ marks (exhaustive search)

	LB0		LB1	LB2		LB3		LB4	
# nodes (level 1)	174	(183 %)	95	95	(100 %)	95	(100 %)	95	(100 %)
# nodes (level 2)	14 854	(172 %)	8 630	8 630	(100 %)	8 630	(100 %)	8 630	(100 %)
# nodes (level 3)	94 131	(107 %)	87 986	87 986	(100 %)	87 986	(100 %)	87 986	(100 %)
# nodes (level 4)	4 662 715	(106 %)	4 409 368	217 768	(5 %)	4 163 250	(94 %)	2 136 616	(48 %)
# nodes (level 5)	16 605 539	(106 %)	15 689 175	404 180	(3 %)	12 962 277	(83 %)	5 781 263	(37 %)
# leafs	12 022 274	(106 %)	11 359 257	246 399	(2 %)	8 878 477	(78 %)	3 724 097	(33 %)
overall # of nodes	33 399 687	(106 %)	31 554 511	965 058	(3 %)	26 100 715	(83 %)	11 738 687	(37 %)
cpu (sec.)	270	(107 %)	253	10	(4 %)	224	(89 %)	143	(56 %)

2-Golomb ruler with $M = 8$ marks (partial search: $0, 4, 24, \ldots$)

	LB0		LB1	LB2		LB3		LB4	
# nodes (level 2)	1	(100 %)	1	1	(100 %)	1	(100 %)	1	(100 %)
# nodes (level 3)	153	(100 %)	153	153	(100 %)	153	(100 %)	153	(100 %)
# nodes (level 4)	18 160	(100 %)	18 160	18 160	(100 %)	17 947	(99 %)	18 160	(100 %)
# nodes (level 5)	1 392 248	(103 %)	1 345 695	1 004 458	(75 %)	1 281 220	(95 %)	663 795	(49 %)
# nodes (level 6)	4 849 696	(103 %)	4 702 898	3 181 092	(68 %)	4 183 570	(89 %)	2 198 371	(47 %)
# leafs	3 475 456	(103 %)	3 375 211	2 195 544	(65 %)	2 920 145	(87 %)	1 552 584	(46 %)
overall # of nodes	9 735 714	(103 %)	9 442 118	6 399 408	(68 %)	8 403 036	(89 %)	4 433 064	(47 %)
cpu (sec.)	135	(103 %)	130	87	(67 %)	122	(94 %)	83	(63 %)

2-Golomb ruler with $M = 11$ marks (partial search: $0, 4, 24, 101, 300, 518, 855, \ldots$)

	LB0		LB1	LB2		LB3		LB4	
# nodes (level 6)	1	(100 %)	1	1	(100 %)	1	(100 %)	1	(100 %)
# nodes (level 7)	1 112	(101 %)	1 100	1 100	(100 %)	1 093	(99 %)	1 100	(100 %)
# nodes (level 8)	145 721	(101 %)	144 299	144 299	(100 %)	141 730	(98 %)	96 365	(67 %)
# nodes (level 9)	397 938	(101 %)	395 124	395 116	(100 %)	385 747	(98 %)	240 957	(61 %)
# leafs	253 326	(101 %)	251 924	251 920	(100 %)	245 109	(97 %)	145 691	(58 %)
overall # of nodes	798 096	(101 %)	792 448	792 436	(100 %)	773 680	(98 %)	484 114	(61 %)
cpu (sec.)	64	(101 %)	64	64	(100 %)	64	(99 %)	46	(73 %)

Table 4: Lower Bound Impact on the Efficiency of the O2GOLRU Algorithm

5 Conclusion

We have generalized the bitmap structures and updating relations used in the best exact algorithms for finding optimum Golomb rulers, leading to new optimum or approximated values for 2-Golomb rulers. Although we have been able to improve the previous results on optimum 2-Golomb rulers, we believe that more efficient algorithms can be built with a better understanding of the structure and relationships between the differences and the

δ-differences. More work needs also to be done on finding sharper lower bounds that are not too expensive to compute although it is still a critical issue for Golomb rulers, see Meyer and Jaumard [7].

We would like to thank D. Haccoun for introducing us to the subject of self doubly orthogonal codes and to the challenging problem of finding optimum 2-Golomb rulers.

References

[1] B. Baechler, *Analyse et détermination de codes doublement orthogonaux pour décodage itératif.* M.Sc. thesis, Ecole Polytechnique de Montréal, Canada, 2000.

[2] B. Baechler, D. Haccoun, and F. Gagnon, "On the Search for Self-Doubly Orthogonal Codes", in *IEEE Int. Symp. Inform. Theory - ISIT 2000 - Conference*, Sorrento, Italy, June 25-30, 2000, 292.

[3] Cardinal, C., D. Haccoun, F. Gagnon, and N. Batani, "Turbo Decoding Using Convolutional Self Doubly Orthogonal Codes", in *ICC '99 - 1999 IEEE International Conference on Communications*, pp. 113-117, 1999.

[4] A. Dollas, W.T. Rankin, and D. McCracken, "A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Ruler", *IEEE Transactions on Information Theory*, Vol. 44, No. 1, pp. 379-382, Jan. 1998.

[5] In Search of the Optimal 20, 21 & 22 Mark Golomb Rulers, http://members.aol.com/golomb20/.

[6] S.C. Hoa, *Construction de règles de Golomb optimales.* M.Sc. thesis, Ecole Polytechnique de Montréal, Canada, 1999.

[7] C. Meyer and B. Jaumard, "Equivalence of some LP-based lower bounds for the Golomb ruler problem", submitted for publication.

[8] Y. Solari, *Les règles de Golomb de degré 2.* Undergraduate Research Project, Ecole Polytechnique de Montréal, Canada, 2003.