Partial reformulation-linearization based optimization models for the Golomb ruler problem

Hacène Ouzia

RAIRO-Oper. Res. 58 (2024) 3171–3188
https://doi.org/10.1051/ro/2024121

RAIRO Operations Research
www.rairo-ro.org

PARTIAL REFORMULATION-LINEARIZATION BASED OPTIMIZATION MODELS FOR THE GOLOMB RULER PROBLEM

Hacène Ouzia*

Abstract. In this paper, we provide a straightforward proof of a conjecture proposed in [P. Duxbury, C. Lavor and L.L. de Salles-Neto, *RAIRO:RO* **55** (2021) 2241–2246.] regarding the optimal solutions of a non-convex mathematical programming model of the Golomb ruler problem. Subsequently, we investigate the computational efficiency of four new binary mixed-integer linear programming models to compute optimal Golomb rulers. These models are derived from a well-known nonlinear integer model proposed in [B. Kocuk and W.-J. van Hoeve, A Computational Comparison of Optimization Methods for the Golomb Ruler Problem. (2019) 409–425.], utilizing the reformulation-linearization technique. Finally, we provide the correct outputs of the greedy heuristic proposed in [P. Duxbury, C. Lavor and L.L. de Salles-Neto, *RAIRO:RO* **55** (2021) 2241–2246.] and correct false conclusions stated or implied therein.

Mathematics Subject Classification. 90C10, 90C11, 90C20, 90C26, 90C27, 90C30.

Received February 26, 2024. Accepted May 31, 2024.

1. Introduction

1.1. The problem

Given a positive integer n, a *ruler* with n marks is a finite increasing sequence $\langle x_0, \dots, x_{n-1}\rangle$ of n integers (the *marks*). The *length* of the ruler $\langle x_0, \dots, x_{n-1}\rangle$ is the difference $x_{n-1} - x_0$. A *Golomb ruler* with n marks is a ruler $\langle x_0, \dots, x_{n-1}\rangle$ such that the inter-distances between its marks are all different. Since the inter-distances are invariant by translation, one can assume that $x_0 = 0$, which implies that x_{n-1} will be the length of the ruler. In the following, we will assume that 0 is the first mark of any Golomb ruler.

The *Golomb Ruler (Optimization) Problem* (GRP) consists of finding, among all Golomb rulers $\langle 0, x_1, \dots, x_{n-1}\rangle$, one that has a minimum length x_{n-1}. A Golomb ruler with minimum length will be called an *optimal Golomb ruler*. For example, in the case of $n = 4$, the two rulers $\langle 0, 1, 4, 6\rangle$ and $\langle 0, 1, 3, 7\rangle$ are Golomb rulers with lengths 6 and 7, respectively. One can easily check that the former is an optimal Golomb ruler among Golomb rulers with 4 marks.

The length of an optimal Golomb ruler with n marks is usually denoted by $G(n)$. To the best of our knowledge, no mathematical expression for $G(n)$ is known, and its values are tabulated only for $n \le 28$. For any given n,

Keywords. Golomb ruler problem, reformulation-linearization techniques, mixed integer linear programming, integer nonlinear programming, quadratic programming.

Sorbonne Université, CNRS, 4 Place Jussieu, Paris 75252, France.
*Corresponding author: hacene.ouzia@sorbonne-universite.fr

the value of $G(n)$ is trivially bounded below by $\binom{n}{2}$ because a Golomb ruler with n marks measures exactly $\binom{n}{2}$ distinct distances [8]. A proof of a better lower bound of $n^2+\sqrt{n}(1-2n)-2$ can be found in [8]. Regarding upper bounds, the best-known upper bound for any number of marks n is $G(n) \leq 2n^3+n$. A conjecture attributed to Erdős states that $G(n) \leq n^2+c$, where $c \in \mathbb{R}$. Computationally, this conjecture was proven to be true for $n \leq 65000$ [8].

The GRP has been studied since the 1960s and remains a very challenging problem from both theoretical and computational points of view. Indeed, to the best of our knowledge, its complexity remains an open question, and it is not even known if the *Golomb Ruler Decision Problem* (given two integers n and ℓ, is there a Golomb ruler with at least n marks and length at most ℓ?) belongs to the complexity class NP or not [23]. Moreover, the optimal Golomb ruler with 28 marks was announced last year after approximately 8.5 years of computational efforts using an enumeration approach implemented on a distributed architecture (`distributed.net` project OGR-28).

During the last few years, in addition to the exact approach implemented in the project OGR-28, two other exact approaches to compute optimal Golomb rulers have also been investigated: constraint programming-based approaches [13] and mathematical programming-based approaches [16, 22]. Moreover, a lot of work has been done to compute near-optimal Golomb rulers using different principles. For instance, construction methods from number theory [10, 11, 26]; hybrid evolutionary heuristic [9]; hybrid search heuristic [28]; and genetic algorithms [36].

Finally, the GRP has many applications. To cite a few, in telecommunication engineering [2], in radio-astronomy [2], in error-correcting codes [30], and x-ray analysis of crystal structures [5].

1.2. Related works

The starting point of this work is the conjecture presented in [12]. This conjecture concerns the solution of the continuous relaxation of a non-convex quadratically constrained integer programming model (see Sect. 3) for the GRP. This model is derived from a mathematical programming model first presented in [16], which we recall below.

Let $\widehat{L}$ be a given upper bound on the length of an optimal Golomb ruler with n marks. The length of an optimal Golomb ruler can be computed using the following nonlinear programming model:

$$(\mathtt{K}_0) \quad \begin{cases} \min & \max\left\{kx_k : k=1,\ldots,\widehat{L}\right\} \\ s.t. & \\ & x_j + \sum_{k=1}^{\widehat{L}-j} x_k x_{k+j} \leq 1, j=1,\ldots,\widehat{L}-1, & \text{(1a)} \\ & x_k \in \{0,1\}, k=1,\ldots,\widehat{L}, & \text{(1b)} \end{cases}$$

where, for every index k, the variable x_k assumes value 1 if and only if the k-th mark is chosen. The j-th quadratic constraint in (1a) imposes that if the j-th mark is selected then no other pair of marks at distance j from each other can be selected.

In [16], the authors conducted a computational comparison between three different optimization approaches to solve the GRP. In the first approach, they considered several enhancements, including bound tightening and branching strategies, to solve two well-known integer linear programming formulations of the GRP [19, 22] using a branch-and-bound algorithm. In the second approach, they considered solving another well-known constraint programming-based model [34, 35] for the GRP and introduced several enhancements as well. In the third approach, they considered solving the GRP using the model ($\mathtt{K}_0$) above. Two approaches based on equivalent formulations of the model ($\mathtt{K}_0$) were considered. The first one is based on a mixed-integer semi-definite reformulation of ($\mathtt{K}_0$). The second is based on what the authors called the *feasibility version* of the model ($\mathtt{K}_0$). This version is used to certify the length of an optimal Golomb ruler (see [16] for more detail). Their approaches differ from the one presented in this work.

Indeed, the approach presented in this work uses the well-known reformulation-linearization approach to derive strong models to the GRP from the following binary mixed integer linear programming (BMILP for short) problem equivalent to the program ($\mathtt{K}_0$). This BMILP model, also stated in [12, 16], reads:

$$(\mathtt{K}_1)\quad\begin{cases} \min & \zeta \\ s.t. & \\ & kx_k \leq \zeta,\ k = 1, \ldots, \widehat{L}, \quad (2a) \\ & \sum_{k=1}^{\widehat{L}} x_k = n-1, \quad (2b) \\ & x_j + \sum_{k=1}^{\widehat{L}-j} x_k x_{k+j} \leq 1, j = 1, \ldots, \widehat{L}-1, \quad (2c) \\ & x_k \in \{0,1\}, k = 1, \ldots, \widehat{L}. \quad (2d) \end{cases}$$

Notice that the equality constraint (2b) is mandatory; otherwise, the zero vector is an optimal solution.

1.3. Our contributions

The contributions of this work are threefold. Firstly, we propose a simple proof of the conjecture stated in [12]. A proof based on arguments from algebraic geometry is presented in the non-peer-reviewed paper [18]. Secondly, we investigate the computational efficiency of four new BMILP models for the GRP. These models are derived from an equivalent BMILP model to the non-linear integer programming model ($\mathtt{K}_0$) using the well-known reformulation-linearization technique. Thirdly, we provide the correct outputs of the greedy heuristic proposed in [12], and we correct false conclusions stated or suggested in therein.

The rest of the paper is organized as follows. In Section 2, we recall the principal concepts and the main theorem of the Reformulation-Linearization technique. Then, in Section 3, we detail our proof of the conjecture stated in [12]. In Section 4, we describe the new BMILP models for the GRP. In Section 5, we investigate the computational efficiency of these proposed models and provide the correct outputs of the greedy heuristic proposed in [12] to compute feasible Golomb rulers. Finally, we give concluding remarks in Section 6.

2. The reformulation linearization technique

The Reformulation Linearization Technique (RLT) for BMILP problems was introduced and studied by Sherali-Adams in [32, 33]. It produces, for a given BMILP model, a finite hierarchy of continuous relaxations with increasing strength, where the relaxation of the higher rank is the description of the convex hull of the set of integer feasible solutions of the considered BMILP. The RLT approach of Sherali-Adams extends the Lift-and-Project hierarchy of Balas [3]. Several other hierarchies are known, such as Lovász-Shrijver [20], Lasserre [17], and the DRL* hierarchy of Minoux and Ouzia [24]. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre hierarchies can be found in [25]. Links between DRL*, RLT, and Lift-and-Project hierarchies can be found in [24]. To be self-contained, we recall below the general RLT principle for BMILP problems; more details can be found in [24, 32, 33].

Let P be a subset of $\mathbb{R}_+^{n_1+n_2}$ describing the set of feasible solutions of a BMILP problem featuring n_1 binary variables and n_2 continuous variables. Let us assume that P features the following *linear description*:

$$\sum_{j=1}^{n_1+n_2} a^j x_j \leq b, \tag{3}$$

$$x_j \leq 1 \text{ for all } j \in E = \{1, \ldots, n_1\}, \tag{4}$$

$$-x_j \leq 0 \text{ for all } j \in N = \{1, \ldots, n_1+n_2\}, \tag{5}$$

$$x_j \in \{0,1\} \text{ for all } j \in E. \tag{6}$$

In the above description, E is the index set of the n_1 binary variables describing P; N is the index set of all the variables; and for each index $j \in N$, the vectors a^j and b belong to $\mathbb{R}^m$, where m is the number of constraints in (3).

The *continuous relaxation* of the integer set P, denoted $\overline{P}$, is the polyhedron defined by the $m + 2n_1 + n_2$ constraints (3)–(5). Recall that two linear descriptions are said to be equivalent if they define the same polyhedron, and a linear description D_1 dominates another linear description D_2 if the polyhedron defined by D_1 is included in the polyhedron defined by D_2.

Let p be a positive integer, and let S be a finite non-empty set. The notation $S^{[p]}$ will denote the set of all subsets of S with cardinality p, while S^p will indicate that the set S has cardinality p (S will be called a p-element set).

Let J^d be a d-element subset of E and let J be a subset from J^d. The *d-factor* associated with the sets J and J^d, denoted $F_d\left(J, J^d \backslash J\right)$, is the degree d polynomial:

$$F_d\left(J, J^d \backslash J\right) = \prod_{j \in J} x_j \prod_{j \in J^d \backslash J} (1 - x_j).$$

We use the convention that $F_0(\emptyset, \emptyset) = 1$. Notice that $F_d\left(J, J^d \backslash J\right)$ is nonnegative for all $x \in [0,1]^{n_1}$.

A *rank d reformulation-linearization relaxation* (of the mixed integer set P) is defined in three steps. First, the problem is *reformulated* as a 0-1 polynomial mixed integer system (semi-algebraic set[2]) by multiplying the constraints (3)–(5) with all d-factors. Then, the nonlinear terms are *linearized* by introducing new variables, giving rise to a higher-dimensional linear description. The last step consists of *projecting* back the resulting polyhedron onto the original x-space. The linearization step can be performed in many different ways, possibly leading to as many different hierarchies of relaxations [24].

The solution set in $\mathbb{R}^{n_1+n_2}$ associated with the nonlinear description resulting from the reformulation step will be denoted R_*^d and reads:

$$R_*^d = \bigcap_{J^d \in E^{[d]}} R^d\left(J^d\right),$$

where, for each subset J^d of E, $R^d\left(J^d\right)$ is the following nonlinear system:

$$\sum_{j=1}^{n_1+n_2} a^j x_j F_d\left(J, J^d \backslash J\right) - b F_d\left(J, J^d \backslash J\right) \leq 0 \quad \text{for all } J \subseteq J^d, \tag{7}$$

$$x_j F_d\left(J, J^d \backslash J\right) - F_d\left(J, J^d \backslash J\right) \leq 0 \quad \text{for all } j \in E \text{ and } J \subseteq J^d, \tag{8}$$

$$x_j F_d\left(J, J^d \backslash J\right) \geq 0 \quad \text{for all } j \in N \text{ and } J \subseteq J^d, \tag{9}$$

$$F_d\left(J, J^d \backslash J\right) \geq 0 \quad \text{for all } J \subseteq J^d. \tag{10}$$

Starting from this non-linear reformulation, various linear descriptions can be constructed depending on the type of linearization considered. Below, we recall the Sherali-Adams linearization [32,33]. Other linearizations are possible [24,27].

The description of the rank d Sherali-Adams relaxation for the polyhedron $\overline{P}$, denoted $\widehat{P}^d_{RLT}$, is a Reformulation-Linearization relaxation of rank d where the nonlinear terms in (7)–(10) are linearized by intro-

[2]A *semi-algebraic* set in n dimensions is a subset of $\mathbb{R}^n$ defined as the solution set of a finite system of polynomial equalities and inequalities; see [4,7].

ducing new variables w_J and w_J^k defined by:

$$w_J = \prod_{j \in J} x_j \text{ for all } J \subseteq E \text{ and } |J| \leq \min(d+1, n_1), \tag{11}$$

$$w_J^k = x_k \prod_{j \in J} x_j \text{ for all } k \in N \backslash E,\ J \subseteq E \text{ and } |J| \leq d, \tag{12}$$

where it is assumed that $w_\emptyset = 1$, and $w_\emptyset^k = x_k$ for every index k in $N \backslash E$.

The resulting higher dimensional linear description will be denoted P_{RLT}^d and it is defined as follows:

$$P_{RLT}^d = \bigcap_{J^d \in E^{[d]}} Q_{RLT}^d\left(J^d\right), \tag{13}$$

where, for each subset J^d of E, the polyhedron $Q_{RLT}^d\left(J^d\right)$ is

$$\sum_{j=1}^{\kappa} a^j W_j^{J,J^d} - bW_0^{J,J^d} \leq 0 \text{ for all } J \subseteq J^d, \tag{14}$$

$$W_j^{J,J^d} - W_0^{J,J^d} \leq 0 \text{ for all } j \in E \text{ and } J \subseteq J^d, \tag{15}$$

$$W_j^{J,J^d} \geq 0 \text{ for all } j \in N \text{ and } J \subseteq J^d, \tag{16}$$

$$W_0^{J,J^d} \geq 0 \text{ for all } J \subseteq J^d. \tag{17}$$

and where, for every j in N, W_j^{J,J^d} and W_0^{J,J^d} denote the linearized forms of the polynomials $x_j F_d\left(J, J^d \backslash J\right)$ and $F_d\left(J, J^d \backslash J\right)$ respectively; these are related to the w_J and w_J^k variables as follows:

$$W_j^{J,J^d} = \sum_{J \subseteq H \subseteq J^d} (-1)^{|H \backslash J|} w_{H \cup \{j\}} \text{ for all } J^d,\ J \subseteq J^d \text{ and } j \in E, \tag{18}$$

$$W_j^{J,J^d} = \sum_{J \subseteq H \subseteq J^d} (-1)^{|H \backslash J|} w_H^j \text{ for all } J^d,\ J \subseteq J^d \text{ and } j \in N \backslash E, \tag{19}$$

$$W_0^{J,J^d} = \sum_{J \subseteq H \subseteq J^d} (-1)^{|H \backslash J|} w_H \text{ for all } J^d,\ J \subseteq J^d. \tag{20}$$

(The above expressions (18)–(20) can also be found in [33]). Notice that the constraints (16)–(17) imply the non-negativity of the variables w_J and w_J^k. The rank d Sherali-Adams relaxation $\widehat{P}_{RLT}^d$ is obtained by projecting the polyhedron P_{RLT}^d onto $\mathbb{R}^{n_1+n_2}$.

The following theorem states the two main results concerning the Sherali-Adams relaxations.

Theorem 2.1 (Sherali-Adams [31,33])**.** *For every integer $d \in \{1, \ldots, n_1 - 1\}$, we have:*

$$\widehat{P}_{RLT}^{d+1} \subseteq \widehat{P}_{RLT}^d.$$

Moreover, the rank n_1 Sherali-Adams relaxation of the set $\overline{P}$ coincides with the convex hull of the integer set P, that is:

$$\mathsf{conv}(P) = \widehat{P}_{RLT}^{n_1}.$$

In essence, this theorem provides a procedure to compute, at least in theory, the convex hull of the integer set P. Additionally, it defines a finite hierarchy of continuous relaxations that can be used as alternatives to the continuous relaxation $\overline{P}$ for optimizing any linear function over P. The rank of each relaxation serves as a measure of its strength.

As a final definition, the Sherali-Adams relaxation obtained using a subset of the d-factors or a subset of the constraints (3) will be called a *partial rank d Sherali-Adams relaxation*. It is evident that any partial rank d Sherali-Adams relaxation contains the corresponding rank d relaxation.

In this work, we will use partial rank 1 and 2 Sherali-Adams relaxations because they feature smaller size. Indeed, after linearizing the nonlinear terms in (7)–(10) using the w variables defined in (11) and (12) above, the P^d_{RLT} description features $\sum\limits_{k=1}^{\min\{d+1,n\}} \binom{n}{k} + m \sum\limits_{k=0}^{d} \binom{n}{k}$ variables (notice that the variable $w_\emptyset$ is not counted here since $w_\emptyset = 1$) and $\mathcal{O}\left(\binom{n}{d}(n_1+n_2)\,2^d\right)$ constraints.

3. Proof of the conjecture

In [12], the authors proposed the following continuous non-convex quadratically constrained model to solve the GRP with n marks:

$$(\widehat{\texttt{K}})\quad \begin{cases} \min & \zeta \\ s.t. & \\ & kx_k \le \zeta x_k,\ k=1,\dots,\widehat{L}, & (21a)\\ & \sum\limits_{k=1}^{\widehat{L}} x_k = n-1, & (21b)\\ & x_j + \sum\limits_{k=1}^{\widehat{L}-j} x_k x_{k+j} \le 1,\ j=1,\dots,\widehat{L}-1, & (21c)\\ & \zeta \in \mathbb{R}_+,\ x_k \in [0,1],\ k=1,\dots,\widehat{L}. & (21d) \end{cases}$$

And, they stated the following conjecture.

Conjecture 3.1. Given an upper bound $\widehat{L}$ of the length of an optimal Golomb rule with n marks. It is true that any optimal solution to the model ($\texttt{K}_1$) is an optimal solution to the model ($\widehat{\texttt{K}}$).

Rephrased differently, this conjecture states that the optimal Golomb rulers with n marks constitute a subset of the set of optimal solutions of the model ($\widehat{\texttt{K}}$).

Before proving this conjecture, one can observe that the model ($\widehat{\texttt{K}}$) is the continuous relaxation of the model ($\texttt{K}$) given below, which in turn is a partial rank 1 Sherali-Adams reformulation of the constraints (2a) of the model ($\texttt{K}_1$).

$$(\texttt{K})\quad \begin{cases} \min & \zeta \\ s.t. & \\ & kx_k \le \zeta x_k,\ k=1,\dots,\widehat{L}, & (22a)\\ & \sum\limits_{k=1}^{\widehat{L}} x_k = n-1, & (22b)\\ & x_j + \sum\limits_{k=1}^{\widehat{L}-j} x_k x_{k+j} \le 1,\ j=1,\dots,\widehat{L}-1, & (22c)\\ & \zeta \in \mathbb{R}_+, x_k \in \{0,1\}, k=1,\dots,\widehat{L}. & (22d) \end{cases}$$

Thus, according to Theorem 2.1, the models ($\texttt{K}_1$) and ($\texttt{K}$) feature the same set of feasible solutions. Indeed, the k-th constraint of (22a) is obtained by first multiplying the k-th constraint of (2a) by the non-negative variable x_k and then replacing x_k^2 with x_k, because x_k is binary. These two steps preserve the feasible solutions of ($\mathtt{K}_1$).

Building upon this observation, in Section 4 and 5, we will propose four new BMILP models for the GRP, derived from the model ($\mathtt{K}_1$) using the partial rank 1 and 2 Sherali-Adams reformulation-linearization technique.

Coming back to the conjecture, below, a straightforward proof.

Proof. (Conjecture 3.1) It is sufficient to prove that any optimal solution to the model ($\mathtt{K}$) is an optimal solution to the model ($\widehat{\mathtt{K}}$). Because, as mentioned above, the two models ($\mathtt{K}$) and ($\mathtt{K}_1$) share the same feasible solutions and thus the same optimal solutions.

Let (L, x_*) be an optimal solution to the model ($\mathtt{K}$). By contradiction, assume that there exists an optimal solution $\left(\widehat{\zeta}, \widehat{x}\right)$ to the model ($\widehat{\mathtt{K}}$) such that $\widehat{\zeta} < L$. Recall that we already have $\widehat{\zeta} \leq L$, because the model ($\widehat{\mathtt{K}}$) is the continuous relaxation of the model ($\mathtt{K}$). It follows from our assumption that, for every feasible solution (ζ, x) of the model ($\widehat{\mathtt{K}}$), it must be true that $x_L = 0$, otherwise, if $x_L \in]0, 1]$ then:

$$Lx_L \leq \zeta x_L \Rightarrow \zeta \geq L, \tag{23}$$

which implies that $\widehat{\zeta} \geq L$, contradicting our assumption. Consequently, the constraint $x_L \leq 0$ is valid for the set of feasible solutions of the model ($\widehat{\mathtt{K}}$) and it cuts off the optimal integer solution (L, x_*) of the model ($\mathtt{K}$). This is impossible because the model ($\widehat{\mathtt{K}}$) contains all the integer solutions of the model ($\mathtt{K}$). Therefore, we must have $\widehat{\zeta} = L$. Since x_* is a feasible solution of the model ($\widehat{\mathtt{K}}$) and it has the same value as its optimal solution $\left(\widehat{\zeta}, \widehat{x}\right)$, then (L, x_*) is an optimal solution of the continuous model ($\widehat{\mathtt{K}}$). This completes the proof. □

Thus, one can state the following:

Theorem 3.2. *Given an upper bound $\widehat{L}$ of the length of an optimal Golomb rule with n marks. It is true that any optimal solution to the model (K_1) is an optimal solution to the model ($\widehat{K}$).*

Two other facts concerning the optimal solutions of the model ($\widehat{\mathtt{K}}$) can be easily demonstrated. Let us emphasize the two parameters n and $\widehat{L}_n$ in the program ($\widehat{\mathtt{K}}$) by the notation ($\widehat{\mathtt{K}}_{n,\widehat{L}_n}$). Let ℓ_n be the length of an optimal Golomb ruler featuring n marks.

Proposition 3.3. *For any optimal solution $\left(\widehat{\zeta}, \widehat{x}\right)$ of the program ($\widehat{K}_{n,\ell_n}$), the component $\widehat{x}_{\ell_n}$ is equal to 1 and $\widehat{\zeta}$ is equal to ℓ_n.*

Proof. Let $\left(\widehat{\zeta}, \widehat{x}\right)$ be an optimal solution the program ($\widehat{\mathtt{K}}_{n,\ell_n}$). Then, $\widehat{\zeta} \leq \ell_n$ and the point $\left(\widehat{\zeta}, \widehat{x}\right)$ satisfies the inequality $\zeta \geq \ell_n x_{\ell_n}$ (obtained by multiplying $x_{\ell_n} \leq 1$ by ζ, which is a positive quantity, and then noticing that $\zeta x_{\ell_n} \geq \ell_n x_{\ell_n}$). Thus, if $\widehat{x}_{\ell_n} < 1$ then $(\ell_n \widehat{x}_{\ell_n}, \widehat{x})$ is a feasible solution to the program ($\widehat{\mathtt{K}}_{n,\ell_n}$) with a better value, contradicting the optimality of $\left(\widehat{\zeta}, \widehat{x}\right)$. Consequently, $\widehat{x}_{\ell_n} = 1$ and $\widehat{\zeta} \geq \ell_n$ implying that $\widehat{\zeta} = \ell_n$. This completes the proof. □

Proposition 3.4. *For any optimal solution $(\ell_n, \widehat{x})$ of the program ($\widehat{K}_{n,\widehat{L}_n}$), the component $\widehat{x}_k = 0$ for $k > \ell_n$.*

Proof. Let $(\ell_n, \widehat{x})$ be an optimal solution to the program ($\widehat{\mathtt{K}}_{n,\widehat{L}_n}$). By contradiction, let us assume that there exists an index $k > \ell_n$ such that $\widehat{x}_k \in]0, 1]$. Thus, we have:

$$\ell_n \widehat{x}_k < k \widehat{x}_k. \tag{24}$$

But, from the constraints (21a), we also have $k\widehat{x}_k \leq \ell_n \widehat{x}_k$. Then, it follows that $\ell_n \widehat{x}_k < \ell_n \widehat{x}_k$, which is absurd. Consequently, for all $k > \ell_n$, we have $\widehat{x}_k = 0$, which completes the proof. □

4. New models

In this section, we provide the descriptions of four new BMILP models proposed to solve the GRP. The first three models are obtained using a partial rank-1 Sherali-Adams relaxation, while the last one is based on a partial rank-2 relaxation.

First, notice that the model ($\mathtt{K}_1$) is equivalent to the following BMILP model:

$$
(\mathtt{G})\quad \begin{cases}
\min \quad \zeta \\
s.t. \\
\quad kx_k \leq \zeta,\ k = 1, \ldots, \widehat{L}, & (25a)\\
\quad \sum_{k=1}^{\widehat{L}} x_k = n-1, & (25b)\\
\quad x_j + \sum_{k=1}^{\widehat{L}-j} w_{k,k+j} \leq 1,\ j = 1, \ldots, \widehat{L}-1, & (25c)\\
\quad x_k + x_j - w_{kj} \leq 1,\ 1 \leq k < j \leq \widehat{L}, & (25d)\\
\quad w_{kj} \leq x_j,\ 1 \leq k < j \leq \widehat{L}, & (25e)\\
\quad w_{kj} \leq x_k,\ 1 \leq k < j \leq \widehat{L}, & (25f)\\
\quad w_{kj} \geq 0,\ 1 \leq k < j \leq \widehat{L}, & (25g)\\
\quad x_k \in \{0,1\}, k = 1, \ldots, \widehat{L}, & (25h)\\
\quad \zeta \in \mathbb{R}_+. & (25i)
\end{cases}
$$

Indeed, the quadratic terms appearing in the constraints (2c) are linearized using the well-known McCormick inequalities [21], which lead to the constraints (25c)–(25f). In the sequel, the constraints (25d)–(25f) will appear frequently. To avoid repeating them, we introduce the following set:

$$
\mathcal{M} = \left\{ (x, w) \in \mathbb{R}^{\widehat{L}} \times \mathbb{R}^{\binom{\widehat{L}}{2}} : \texttt{(25d)--(25g)} \right\}. \tag{26}
$$

Linearizing in the model ($\mathtt{K}$) the products ζx_k using new variables v_k and using the McCormick inequalities to linearize the quadratic terms appearing in the constraints (22c), we obtain the following BMILP equivalent model:

$$
(\mathtt{M}_1)\quad \begin{cases}
\min \quad \zeta \\
s.t. \\
\quad kx_k \leq v_k,\ k = 1, \ldots, \widehat{L}, & (27a)\\
\quad v_k \leq \zeta,\ k = 1, \ldots, \widehat{L}, & (27b)\\
\quad v_k \leq \widehat{L} x_k,\ k = 1, \ldots, \widehat{L}, & (27c)\\
\quad v_k \geq \zeta + \widehat{L}\left(x_k - 1\right),\ k = 1, \ldots, \widehat{L}, & (27d)\\
\quad \sum_{k=1}^{\widehat{L}} x_k = n-1, & (27e)\\
\quad x_j + \sum_{k=1}^{\widehat{L}-j} w_{k,k+j} \leq 1,\ j = 1, \ldots, \widehat{L}-1, & (27f)\\
\quad \zeta \in \mathbb{R}_+, (x, w) \in \mathcal{M}, & (27g)\\
\quad x_k \in \{0,1\}, k = 1, \ldots, \widehat{L}. & (27h)
\end{cases}
$$

The model ($\mathtt{M}_1$) can be seen as derived from the model ($\mathtt{G}$) using a partial rank 1 Sherali-Adams reformulation, where the k-th constraint (25a) is first multiplied by x_k. Then, the term x_k^2 is replaced by x_k, because x_k is binary. Finally, the products ζx_k are linearized using the constraints (27a)–(27d). To avoid repeating the linearization constraints (27b)–(27d), let $\mathcal{V}$ be the following set:

$$\mathcal{V} = \left\{ (x, w, v) \in \mathbb{R}^{\widehat{L}} \times \mathbb{R}^{\binom{\widehat{L}}{2}} \times \mathbb{R}^{\widehat{L}} : \text{(27b)--(27d)} \right\}. \tag{28}$$

In the sequel, we propose two other BMILP models for the GRP. These models are obtained, as above, by a partial rank 1 Sherali-Adams reformulation-linearization approach applied to the model ($\mathtt{G}$).

The model ($\mathtt{M}_2$) below is obtained from the model ($\mathtt{G}$) using a complete rank 1 Sherali-Adams reformulation-linearization of the constraints (25a). This means that these constraints are first multiplied by the 1-factors x_j and $1 - x_j$, for all $j \in \left\{1, \dots, \widehat{L}\right\}$. Then, the products ζx_k are linearized using the new variables v_k. Additionally, for every pair of indexes (i, j) such that $i < j$, the product $x_i x_j$ is linearized using the variable w_{ij}. This model ($\mathtt{M}_2$) reads:

$$(\mathtt{M}_2) \quad \left\{ \begin{array}{lll} \min & \zeta & \\ s.t. & & \\ & k x_k \le v_k, \ k = 1, \dots, \widehat{L}, & \text{(29a)} \\ & k w_{kj} \le v_j, \ 1 \le k < j \le \widehat{L}, & \text{(29b)} \\ & k x_k - k w_{kj} \le \zeta - v_j, \ 1 \le k < j \le \widehat{L}, & \text{(29c)} \\ & \sum_{k=1}^{\widehat{L}} x_k = n - 1, & \text{(29d)} \\ & x_j + \sum_{k=1}^{\widehat{L}-j} w_{k,k+j} \le 1, \ j = 1, \dots, \widehat{L} - 1, & \text{(29e)} \\ & \zeta \in \mathbb{R}_+, (x, w) \in \mathcal{M}, (x, w, v) \in \mathcal{V}, & \text{(29f)} \\ & x_k \in \{0, 1\}, k = 1, \dots, \widehat{L}. & \text{(29g)} \end{array} \right.$$

The model ($\mathtt{M}_3$) below is obtained from the model ($\mathtt{G}$) using a complete rank 1 Sherali-Adams reformulation-linearization of the constraints (25a) and (25b). In other words, the model ($\mathtt{M}_3$) is obtained from the model ($\mathtt{M}_2$) using, in addition, a complete rank 1 Sherali-Adams reformulation-linearization of the constraint (29d), where only the 1-factors $\left\{x_j : j = 1, \dots, \widehat{L}\right\}$ are needed. We obtain the same constraints using the 1-factors $\left\{1 - x_j : j = 1, \dots, \widehat{L}\right\}$.

$$
(\mathtt{M}_3)\quad
\begin{cases}
\min & \zeta \\
s.t. & \\
& kx_k \leq v_k,\ k = 1, \ldots, \widehat{L}, \qquad (30a)\\
& kw_{kj} \leq v_j,\ 1 \leq k < j \leq \widehat{L}, \qquad (30b)\\
& kx_k - kw_{kj} \leq \zeta - v_j,\ 1 \leq k < j \leq \widehat{L}, \qquad (30c)\\
& \sum_{k=1}^{\widehat{L}} x_k = n - 1, \qquad (30d)\\
& \sum_{k=1}^{j-1} w_{kj} + \sum_{k=j+1}^{\widehat{L}} w_{jk} = (n-2)\, x_j,\ j = 1, \ldots, \widehat{L} \qquad (30e)\\
& x_j + \sum_{k=1}^{\widehat{L}-j} w_{k,k+j} \leq 1,\ j = 1, \ldots, \widehat{L} - 1, \qquad (30f)\\
& \zeta \in \mathbb{R}_+, (x, w) \in \mathcal{M}, (x, w, v) \in \mathcal{V}, \qquad (30g)\\
& x_k \in \{0, 1\}, k = 1, \ldots, \widehat{L}. \qquad (30h)
\end{cases}
$$

Finally, we introduce the model ($\mathtt{M}_4$), which is obtained through a partial rank-2 Sherali-Adams reformulation of the base model ($\mathtt{G}$) as outlined below.

$$
(\mathtt{M}_4)\quad
\begin{cases}
\min & \zeta \\
s.t. & \\
& kx_k \leq v_k,\ 1 \leq k \leq \widehat{L}, \qquad (31a)\\
& kw_{kj} \leq u_{kj},\ 1 \leq k < j \leq \widehat{L}, \qquad (31b)\\
& kw_{kj} - kx_k \leq v_k - u_{kj},\ 1 \leq k < j \leq \widehat{L}, \qquad (31c)\\
& u_{kj} \leq \zeta,\ 1 \leq k < j \leq \widehat{L}, \qquad (31d)\\
& u_{kj} \leq \widehat{L} w_{kj},\ 1 \leq k < j \leq \widehat{L}, \qquad (31e)\\
& u_{kj} \geq \zeta + \widehat{L}\,(w_{kj} - 1)\,,\ 1 \leq k < j \leq \widehat{L}, \qquad (31f)\\
& \sum_{k=1}^{\widehat{L}} x_k = n - 1, \qquad (31g)\\
& x_j + \sum_{k=1}^{\widehat{L}-j} w_{k,k+j} \leq 1,\ j = 1, \ldots, \widehat{L} - 1, \qquad (31h)\\
& \zeta \in \mathbb{R}_+, (x, w) \in \mathcal{M}, (x, w, v) \in \mathcal{V}, \qquad (31i)\\
& u_{kj} \geq 0,\ 1 \leq k < j \leq \widehat{L}, \qquad (31j)\\
& x_k \in \{0, 1\}, k = 1, \ldots, \widehat{L}. \qquad (31k)
\end{cases}
$$

The constraints (25a) of the model ($\mathtt{G}$) are reformulated using the factors:

$$
\left\{x_k : 1 \leq k \leq \widehat{L}\right\} \text{ and } \left\{x_k x_j, x_k\,(1 - x_j) : 1 \leq k < j \leq \widehat{L}\right\}. \qquad (32)
$$

The new variables u_{kj} are used to linearize the new products $\zeta x_k x_j$. The variables v and w are as defined previously.

5. Computational results

The computational results presented in this section have two purposes. The first one is to provide the correct outputs of the greedy heuristic presented in [12] for computing feasible Golomb rulers. The second purpose is to compare the computational efficiency of the models (G), (M_1), (M_2), (M_3), (M_4), and ($\widehat{K}$) in solving the GRP. Moreover, we also correct the false conclusions stated or implied in [12].

5.1. The heuristic

The computational results reported in this subsection were obtained using a Dell-Optiplex desktop with an Intel-Core-i7-9700 CPU running at 2.0 GHz with 8 cores, and 32 GB of RAM. The desktop is operated by a Linux Ubuntu 18.05.06 LTS operating system.

In [12], the authors proposed a greedy heuristic (referred to as H) to compute feasible Golomb rulers. For the sake of completeness, we provide its pseudo-code below. The variables mark and dist are two boolean arrays such that mark(k) $= 1$ (resp. dist(k) $= 1$) if and only if the mark (resp. distance) k is not used.

Algorithm 1: Greedy heuristic H.

```
Input n (number of marks)
Outputs ruler (set containing ruler's marks) ; lruler (ruler's length)
begin
    limit ← MaxInt                                  // Upper-bound to the Golomb ruler's length
    for k ∈ {0, 1} do
        mark(k) ← 0; dist(k) ← 0
    for k ∈ {2, ..., limit} do
        mark(k) ← 1; dist(k) ← 1
    lruler ← 1; k ← 1
    while k < n − 1 do
        d ← 1; infeasible ← 1
        while infeasible do
            if dist(d) then
                length ← lruler + d
                i ← 0                               // Check if length is feasible.
                while (mark(i) or dist(length − i)) and i < length + 1 do
                    i ← i + 1
                if i = length + 1 then
                    lruler ← length
                    mark(lruler) ← 0
                    for i ∈ {0, ..., lruler − 1} do
                        if not mark(i) then dist(lruler − i) ← 0
                    infeasible ← 0
            d ← d + 1
        k ← k + 1
    ruler ← {}                                      // Extract ruler's marks
    for k ∈ {0, ..., lruler} do
        if not mark(k) then ruler ← ruler ∪ {k}
    return ruler, lruler
```

TABLE 1. Correct lengths obtained using the greedy heuristic `H`.

n	Length		Gap		n^2
	`Opt`	`H`	`abs.`	`rel.(%)`	
5	11	12	1	8.33	25
6	17	20	3	15.00	36
7	25	30	5	16.67	49
8	34	44	10	22.72	64
9	44	65	21	32.31	81
10	55	80	25	31.25	100
11	72	96	24	25.00	121
12	85	122	37	30.33	144
13	106	147	41	27.89	169
14	127	181	54	29.83	196
15	151	203	52	25.62	225
16	177	251	74	29.48	256
17	199	289	90	31.14	289
18	216	360	144	40.00	324
19	246	400	154	38.50	361
20	283	474	191	40.30	400
21	333	564	231	40.96	441
22	356	592	236	39.86	484
23	372	661	289	43.72	529
24	425	774	349	45.09	576
25	480	821	341	41.53	625
26	492	915	423	46.23	676
27	553	969	416	42.93	729
28	585	1015	430	42.36	784

Unfortunately, the implementation proposed in https://github.com/luizleduino/golombruler/blob/master/heuristic is buggy and the lengths reported in [12] are incorrect. Indeed, as implemented, the heuristic `H` returns non-Golomb rulers for $n \geq 9$. In the case of $n = 9$, it returns the ruler $\langle 0, 1, 3, 7, 12, 20, 30, 44, 59 \rangle$, which is not a Golomb ruler because $59 - 30$ is equal to $30 - 1$. Additionally, the computed rulers with $n \geq 10$ all have as their first 10 marks the marks of the ruler above. Thus, all these rulers are not Golomb rulers.

The correct lengths of the Golomb ruler computed by the heuristic `H` are shown in Table 1, where for a given integer n, the entry `Opt`(n) in the column `Opt` (resp. `H`(n) in the column `H`) is the length of an optimal Golomb ruler (resp. the length of the Golomb ruler computed by the heuristic `H`) featuring n marks. The meaning of the other columns is straightforward. From the aforementioned table, one can observe that for $n \geq 17$, the difference between n^2 and lengths `H`(n) tends to increase, contrary to the computational results reported in Table 1 of [12]. Thus, the conclusion drawn in [12] regarding the good performances of the heuristic `H` to compute Golomb rulers (see Sect. 5, second paragraph) is wrong.

5.2. Numerical efficiency of the proposed models

The computational results presented in this subsection aim to investigate the numerical efficiency of, first, the model ($\widehat{\mathtt{K}}$), and then the models ($\mathtt{G}$), ($\mathtt{M}_1$), ($\mathtt{M}_2$), ($\mathtt{M}_3$), and ($\mathtt{M}_4$) to solve the `GRP`. These computational results were obtained using an `HPC cluster` featuring `2xIntel-Xeon-5690` processors with `12 cores` and `24 threads`. The solution time was limited to 24 hours. All the `BMILP` models ($\mathtt{G}$), ($\mathtt{M}_1$), ($\mathtt{M}_2$), ($\mathtt{M}_3$), and ($\mathtt{M}_4$) were solved using the solver `Gurobi` (`ver-10.0.1`) [14] with its default settings, while the solvers `Baron` (`ver-23.3.11`) [15] and `Knitro` (`ver-13.2.1`) [6] were used to solve the model ($\widehat{\mathtt{K}}$). These solvers were all used *via* the `AMPL` interface [1, 29].

TABLE 2. Computational efficiency of the model ($\widehat{\mathtt{K}}$) to solve the GRP.

			Size		Knitro		Baron			
n	Opt	$\widehat{L}$	nv	nc	Objval	Time (m:s)	Lb	Ub	Gap (%)	Time (h:m:s)
5	11	12	13	36	11.00	4.08	11.00	11	0.00	10.92
6	17	20	21	60	17.00	8.41	14.46	17	14.94	24:06:55
7	25	30	31	90	25.00	17.69	15.34	25	38.62	24:00:05
8	34	44	45	132	34.00	39.63	14.74	37	60.18	24:00:06
9	44	65	66	195	45.00	1:10	15.47	49	68.44	24:00:07
10	55	80	81	240	62.00	2:09	17.48	63	72.26	24:00:02
11	72	96	97	288	78.00	3:45	18.55	79	76.52	24:00:03
12	85	122	123	366	94.00	8:03	20.06	106	81.08	24:00:02
13	106	147	148	441	113.00	13:32	21.12	123	82.83	24:00:01
14	127	181	182	543	133.00	28:12	22.09	147	84.97	24:00:09

Firstly, in [12], the authors advocated that to solve the GRP, it is more promising to solve the continuous relaxation ($\widehat{\mathtt{K}}$) than solving the model (K) (see Sect. 5, first paragraph). This is rather counterintuitive because the nonlinear model ($\widehat{\mathtt{K}}$) is non-convex. As an argument, they showed computational results using the solver Knitro with a particular setting (multi-start enabled with 5000 random points).

In Table 2, we reproduced the computational results obtained after solving the non-convex model ($\widehat{\mathtt{K}}$) using the two solvers Knitro (with the same setting used in [12]) and Baron (with its default settings). For the Knitro solver, we reported the solution time in seconds (column Time) and the objective value (column Objval). For the Baron solver, we reported the lower bounds (column Lb), the upper bounds (column Ub), the relative gap (column Gap) between the upper and lower bounds, and the solution time in a human-readable format (column Time). The meaning of the other columns is as follows: the columns n, Opt, $\widehat{L}$ are respectively the number of marks, the length of an optimal Golomb rulers, and the used upper bound to the length of an optimal Golomb ruler. The content of the two columns nv and nc are respectively the number of variables and constraints of the model ($\widehat{\mathtt{K}}$).

The solver Knitro, as observed in [12], solves the model ($\widehat{\mathtt{K}}$) quickly. However, it returns only feasible solutions, as indicated in the column Objval. This is not surprising because Knitro guarantees a global optimal solution only if the model is convex, which is not the case for the model ($\widehat{\mathtt{K}}$). In contrast to the Knitro solver, the Baron solver guarantees a global optimal solution of the model ($\widehat{\mathtt{K}}$). It is notable that the relative gap after 24 hours (time limit) of computational efforts is at least 70% for instances with a number of marks $n \geq 10$. As we will demonstrate latter, better performances can be obtained using BMILP models to solve the GRP, contrary to what is suggested in [12] (see Sect. 5, first paragraph).

Secondly, let us compare the relative performance of the models (G), ($\mathtt{M}_1$), ($\mathtt{M}_2$), and ($\mathtt{M}_3$) to solve the GRP. The results of our computational experiments are compiled in Table 3, where for each instance, we reported the number of marks (column n), the length of an optimal Golomb ruler (column Opt), and the upper bound to the length of an optimal Golomb ruler used (column $\widehat{L}$). For each model, we report its size: number of binary and continuous variables (columns nbv and ncv, respectively); total number of variables and constraints (columns nv and nc, respectively); the value of the best bound (column Best bnd.); the value of the best incumbent found (column Best sol.); the relative gap computed as $1 - \frac{\texttt{Best bnd.}}{\texttt{Best sol.}}$ (column Gap); the solution time (column Time); and the number of explored nodes in the branch-and-bound tree (column Nbr. nodes).

From Table 3, one can observe that the model ($\mathtt{M}_1$) outperforms the other three models if one considers the number of optimal Golomb rulers found (the number of optimal Golomb rulers found by the models ($\mathtt{M}_1$), (G), ($\mathtt{M}_2$), and ($\mathtt{M}_3$) are $8, 7, 6, 6$, respectively), the values of the best lower bounds (except for the ruler with 13 marks for which the lower bound computed by (G) is better), or the running time (except for the ruler with 11 marks

TABLE 3. Computational performance of the models $\mathtt{G}$, $\mathtt{M}_1, \mathtt{M}_2$, and $\mathtt{M}_3$ to solve the GRP.

			Model G										Model M_1									
			Size of the model				Best	Best	Gap	Time	Nbr.	Size of the model				Best	Best	Gap	Time	Nbr.
n	Opt.	$\widehat{L}$	nbv	ncv	nv	nc	bnd.	sol.	(%)	(h:m:s)	Nodes	nbv	ncv	nv	nc	bnd.	sol.	(%)	(h:m:s)	Nodes
5	11	12	12	67	79	222	11.00	11	0.00	0.25	1	12	79	91	258	11.00	11	0.00	0.28	1
6	17	20	20	191	211	610	17.00	17	0.00	0.37	1	20	211	231	670	17.00	17	0.00	0.26	1
7	25	30	30	436	466	1365	25.00	24	0.00	1.9	193	30	466	496	1455	25.00	25	0.00	2.46	1
8	34	44	44	947	991	2926	34.00	34	0.00	9.31	1893	44	991	1035	3058	34.00	34	0.00	16.4	2277
9	44	65	65	2081	2146	6370	44.00	44	0.00	1:03	12481	65	2146	2211	6565	44.00	43	0.00	1:15	9555
10	55	80	80	3161	3241	9640	55.00	55	0.00	12:24	35994	80	3241	3321	9880	55.00	55	0.00	7:46	35817
11	72	96	96	4561	4657	13872	72.00	72	0.00	1:39:34	1593120	96	4657	4753	14160	72.00	72	0.00	3:32:15	1241720
12	85	122	122	7382	7504	22387	76.00	94	19.15	limit	18006203	122	7504	7626	22753	85.00	85	0.00	15:24:35	4815450
13	106	147	147	10732	10879	32487	94.00	115	18.26	limit	14230945	147	10879	11026	32928	71.00	117	39.32	limit	485739
14	127	181	181	16291	16472	49232	75.32	140	46.20	limit	987556	181	16472	16653	49775	84.00	143	41.26	limit	757324
			Model M_2									Model M_3								
			Size of the model				Best	Best	Gap	Time	Nbr.	Size of the model				Best	Best	Gap	Time	Nbr.
n	Opt.	$\widehat{L}$	nbv	ncv	nv	nc	bnd.	sol.	(%)	(h:m:s)	Nodes	nbv	ncv	nv	nc	bnd.	sol.	(%)	(h:m:s)	Nodes
5	11	12	12	145	157	588	11.00	11	0.00	0.34	1	12	145	157	600	11.00	11	0.00	0.31	1
6	17	20	20	401	421	1620	17.00	17	0.00	0.7	1	20	401	421	1640	17.00	17	0.00	0.78	1
7	25	30	30	901	931	3630	25.00	25	0.00	4.55	1	30	901	931	3660	25.00	25	0.00	8.55	221
8	34	44	44	1937	1981	7788	34.00	34	0.00	35.12	2031	44	1937	1981	7832	34.00	34	0.00	23.95	1988
9	44	65	65	4226	4291	16965	44.00	44	0.00	3:21	12244	65	4226	4291	17030	44.00	43	0.00	2:45	4239480
10	55	80	80	6401	6481	25680	55.00	55	0.00	42:53	35195100	80	6401	6481	25760	55.00	55	0.00	44:13	2840990
11	72	96	96	9217	9313	36960	71.00	74	4.05	limit	439260	96	9217	9313	37056	65.00	72	9.72	limit	241217
12	85	122	122	14885	15007	59658	63.99	95	32.64	limit	186046	122	14885	15007	59780	64.31	95	32.31	limit	131492
13	106	147	147	21610	21757	86583	61.56	122	49.54	limit	70359	147	21610	21757	86730	67.45	123	45.16	limit	49950
14	127	181	181	32762	32943	131225	52.75	153	65.52	limit	18043	181	32762	32943	131406	47.07	147	67.98	limit	26898

for which the running time to compute the optimal Golomb ruler by the model (G) is approximately half the running time of the model (M_1)).

When examining specific pairs of models, first, one can notice that the two models (K) and (M_1) perform rather similarly on rulers featuring a number of marks between 5 and 11. For the other rulers, the model (M_1) performs better than the model (K). Indeed, an optimal Golomb ruler with 12 marks is found in less than 16 hours using the model (M_1), while the relative gap of the feasible Golomb ruler found using the model (K) after 24 hours of computational efforts is 19.15%. For the ruler with 13 marks, the relative gap obtained using the model (K) is better than the one obtained using the model (M_1). However, a slightly better relative gap using the model (M_1) is obtained for the ruler with 14 marks. Second, the performances of the two models (M_2) and (M_3) are rather similar. Optimal Golomb rulers are obtained for the GRP with a number of marks between 5 and 10. For rulers with a number of marks between 11 and 14, the relative gaps are slightly the same.

Regarding the number of nodes explored during the branch-and-bound algorithm, the model (K) consistently exhibits the highest counts across almost all instances. In contrast, the number of nodes explored using the model (M_1) surpasses those of models (M_2) and (M_3). Comparing the number of nodes explored by the latter two models is more complex. However, the reduced count of explored nodes in models (M_2) and (M_3) compared to (K) and (M_1) can be attributed to the size (number of variables and constraints) of their continuous relaxations. This indicates that solving the continuous relaxation at each node of the branch-and-bound algorithm is more time-consuming.

Based on the aforementioned analysis, it appears that the model (M_1) exhibits superior performance in computing optimal Golomb rulers with a number of marks ranging from 5 to 14.

At this point of the discussion, one may wonder about the efficiency of the model (M_1) to solve instances with a number of marks $n \geq 15$. Moreover, one may ask about using a rank 2 partial Sherali-Adams relaxation.

In Table 4, we present computational results showing the performances of the two models, (M_1) and (M_4), in computing optimal Golomb rulers when the number of marks n ranges from 13 to 18. The computation time was limited to 10 days.

One can observe that computing optimal Golomb rulers for instances featuring a number of marks n between 13 and 18 is time-consuming. Even after 10 days of computational efforts, the relative gap remains high for instances with 14 marks or more.

TABLE 4. Computational performance of the models $\mathtt{M}_1$ and $\mathtt{M}_4$.

n	Opt.	$\widehat{L}$		Size of the model				Best	Best	Gap	Time	Nbr.
				nbv	ncv	nv	nc	bnd.	sol.	(%)	(h:m:s)	Nodes
			Model $\mathtt{M}_1$									
13	106	147		147	10879	11026	32928	98.00	109	10.09	240:00:01	21738900
14	127	181		181	16472	16653	49775	99.00	138	28.26	240:00:10	23501800
15	151	203		203	20707	20910	62524	93.52	170	44.99	240:00:04	2368570
16	177	251		251	31627	31878	95380	100.00	204	50.98	240:00:03	696569
17	199	289		289	41906	42195	126293	78.39	242	67.61	240:00:05	550219
18	216	360		360	64981	65341	195660	52.64	287	81.72	240:00:04	45491
			Model $\mathtt{M}_4$									
13	106	147		147	21610	21757	86583	74.00	119	37.82	240:00:03	1898050
14	127	181		181	32762	32943	131225	69.50	148	53.04	240:00:05	1162260
15	151	203		203	41210	41413	165039	83.00	175	52.84	240:00:04	286779
16	177	251		251	63002	63253	252255	62.24	209	70.22	240:00:05	45
17	199	289		289	83522	83811	334373	80.22	256	68.67	240:00:06	38518
18	216	360		360	129601	129961	518760	85.54	293	70.81	240:00:09	47880

Additionally, the results obtained using the model ($\mathtt{M}_1$) are competitive with those obtained by the model ($\mathtt{M}_4$). The model ($\mathtt{M}_1$) finds Golomb rulers with better lengths compared to those found by the model ($\mathtt{M}_4$). Also, except for the two last rulers, the values of the lower bounds computed by the model ($\mathtt{M}_1$) are better than those computed by the model ($\mathtt{M}_4$). Regarding the number of explored nodes during the branch-and-bound algorithm, the model ($\mathtt{M}_1$) explores more nodes than the model ($\mathtt{M}_4$), because the latter features a continuous relaxation that is stronger but time-consuming to solve.

To gain a better understanding of the time necessary to compute Golomb rulers using the models ($\mathtt{M}_1$) and ($\mathtt{M}_4$), we fitted the values of the relative gap for rulers with $n \in \{16, 17, 18\}$ using the model:

$$g_n(t) = 1 - \frac{1}{1 + \alpha_n t^{-\beta_n}}, \tag{33}$$

where t is the time variable in days.

The values of the parameters α_n and β_n are estimated based on the relative gaps recorded in Table 5. These relative gaps are those recorded after each day of computational effort during the 10 days. The estimated values of the parameters α_n and β_n are given in the last two rows.

In Table 6, we reported the estimated time necessary to compute (with the same resources) Golomb rulers with a number of marks $n \in \{16, 17, 18\}$ and featuring a relative gap at most the value indicated in the first column. For instance, using the model ($\mathtt{M}_1$), almost 8.5 years are necessary to obtain a Golomb ruler with 16 marks featuring a relative gap of at most 10%. One can observe that the estimated time increases drastically with the number of marks and the desired value of the relative gap. Computing Golomb rulers with 18 marks and a relative gap of at most 50% is out of reach using the model ($\mathtt{M}_1$). In contrast, there is a better hope using the model ($\mathtt{M}_4$), with at most 2 years to compute a Golomb ruler with a relative gap of at most 50%.

6. Conclusion

In this work, we proposed a straightforward proof of the conjecture stated in [12] regarding the optimal solutions of the model ($\widehat{\mathtt{K}}$). Moreover, concerning the greedy heuristic proposed in [12] to compute Golomb rulers, we provided its correct outputs and corrected the erroneous facts and conclusions presented in [12].

As a second contribution, starting from the observation that the proposed model ($\widehat{\mathtt{K}}$) is the continuous relaxation of the model ($\mathtt{K}$), derived from the model ($\mathtt{G}$) using the well-known RLT technique, we explored the computational efficiency of four models, all obtained by partial Sherali-Adams reformulation-linearization

TABLE 5. Values of the relative gaps recorded each day of the computational experiment and the estimated values of the parameters α_n and β_n.

	Instances					
	Solved using M_1			Solved using M_4		
Day	16	17	18	16	17	18
1	0.798	0.804	0.822	0.761	0.786	0.788
2	0.542	0.803	0.822	0.735	0.708	0.775
3	0.536	0.773	0.822	0.722	0.707	0.737
4	0.529	0.692	0.822	0.702	0.698	0.731
5	0.524	0.684	0.822	0.702	0.696	0.724
6	0.523	0.680	0.821	0.702	0.687	0.723
7	0.522	0.679	0.821	0.702	0.687	0.710
8	0.522	0.678	0.821	0.702	0.687	0.708
9	0.522	0.677	0.820	0.702	0.687	0.708
10	0.510	0.676	0.817	0.702	0.687	0.708
α_n	2.16449	4.34312	4.652	3.01332	3.08277	3.68457
β_n	0.369887	0.355974	0.0093652	0.126109	0.173105	0.198012

TABLE 6. Estimated solution time in years using the models (M_1) and (M_4) to compute Golomb rulers with number of marks $n \in \{16, 17, 18\}$.

	Number of marks		
Targeted gap (%)	16	17	18
Model M_1			
50	0.022098	0.169597	5.33463×10^{68}
25	0.430789	3.71317	4.713×10^{119}
10	8.398	81.2966	4.1638×10^{170}
5	63.3142	663.266	1.86307×10^{205}
Model M_4			
50	17.2329	1.8291	1.98642
25	104652	1043.5	510.067
10	6.35531×10^{8}	595312	130973
5	2.37895×10^{11}	4.4606×10^{7}	5.70199×10^{6}

technique. The computational results we provided indicate, among other things, that using BMILP models is more efficient than non-convex models, for obvious reasons. Also, they indicate that computing optimal Golomb rulers when $n \geq 13$ is time-consuming. This is not surprising if one recalls that the optimal Golomb Ruler with 28 marks was obtained after approximately 8.5 years of computational time. Perhaps an equivalent or greater amount of time is required to compute optimal Golomb rulers using Reformulation-Linearization-based models.

ACKNOWLEDGEMENTS

We are grateful to the referees for their careful reading and insightful, precise comments.

References

[1] AMPL Optimization Inc, AMPL. https://ampl.com (2023).

[2] W.C. Babcock, Intermodulation interference in radio systems frequency of occurrence and control by channel selection. *Bell Syst. Tech. J.* **32** (1953) 63–73.

[3] E. Balas, S. Ceria and G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Math. Program.* **58** (1993) 295–324.

[4] S. Basu, R. Pollack and M.-F. Roy, Algorithms in Real Algebraic Geometry. Springer Berlin Heidelberg (2006).

[5] G. Bloom and S. Golomb, Applications of numbered undirected graphs. *Proc. IEEE* **65** (1977) 562–570.

[6] R.H. Byrd, J. Nocedal and R.A. Waltz, Knitro: An Integrated Package for Nonlinear Optimization. (2006) 35–59.

[7] D. Cox, J. Little and D. O'Shea, Using Algebraic Geometry, 2nd edition. Springer (2004).

[8] A. Dimitromanolakis, *Analysis of the golomb ruler and the sidon set problems, and determination of large, near-optimal golomb rulers*. Master's thesis, Department of Electronic and Computer Engineering, Technical University of Crete (2002).

[9] I. Dotú and P. Van Hentenryck, A simple hybrid evolutionary algorithm for finding golomb rulers. In Vol. 3 *2005 IEEE Congress on Evolutionary Computation*. IEEE (2005) 2018–2023.

[10] K. Drakakis, A review of the available construction methods for golomb rulers. *Adv. Math. Commun.* **3** (2009) 235–250.

[11] K. Drakakis, R. Gow and L. O'Carroll, On some properties of costas arrays generated via finite fields. In: *2006 40th Annual Conference on Information Sciences and Systems*. IEEE (2006) 801–805.

[12] P. Duxbury, C. Lavor and L.L. de Salles-Neto, A conjecture on a continuous optimization model for the golomb ruler problem. *RAIRO:RO* **55** (2021) 2241–2246.

[13] P. Galinier, A Constraint-Based Approach to the Golomb Ruler Problem. Université de Montréal, Centre de recherche sur les transports (2003).

[14] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual. https://www.gurobi.com (2023).

[15] A. Khajavirad and N.V. Sahinidis, A hybrid lp/nlp paradigm for global optimization relaxations. *Math. Program. Comput.* **10** (2018) 383–421.

[16] B. Kocuk and W.-J. van Hoeve, A Computational Comparison of Optimization Methods for the Golomb Ruler Problem. (2019) 409–425.

[17] J.B. Lasserre, An Explicit Exact sdp Relaxation for Nonlinear 0–1 Programs. (2001) 293–303.

[18] T. Liu and C. Luo, The proof of a conjecture for a continuos golumb ruler model. DOI: https://doi.org/10.20944/preprints202211.0027.v2 (2022).

[19] R. Lorentzen and R. Nilsen, Application of linear programming to the optimal difference triangle set problem. *IEEE Trans. Inf. Theory* **37** (1991) 1486–1488.

[20] L. Lovász and A. Schrijver, Cones of matrices and set-functions and 0–1 optimization. *SIAM J. Optim.* **1** (1991) 166–190.

[21] G.P. McCormick, Computability of global solutions to factorable nonconvex programs: part i - convex underestimating problems. *Math. Program.* **10** (1976) 147–175.

[22] C. Meyer and B. Jaumard, Equivalence of some lp-based lower bounds for the golomb ruler problem. *Discrete Appl. Math.* **154** (2006) 120–144.

[23] C. Meyer and P.A. Papakonstantinou, On the complexity of constructing golomb rulers. *Discrete Appl. Math.* **157** (2009) 738–748.

[24] M. Minoux and H. Ouzia, DRL*: A hierarchy of strong block-decomposable linear relaxations for 0–1 mips. *Discrete Appl. Math.* **158** (2010) 2031–2048.

[25] L. Monique, A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0–1 programming. *Math. Oper. Res.* **28** (2003) 470–496.

[26] C.A.M. Ojeda, D.F.D. Urbano and C.A.T. Solarte, Near-optimal g-golomb rulers. *IEEE Access* **9** (2021) 65482–65489.

[27] H. Ouzia, Two new reformulation convexification based hierarchies for 0–1 mips. *Adv. Oper. Res.* (2015) (2015) 1–13.

[28] S. Prestwich, Trading completeness for scalability: Hybrid search for cliques and rulers. In: *Proceedings of the Third International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. (2001) 159–174.

[29] D.M.G. Robert Fourer and B.W. Kernighan, A modeling language for mathematical programming. *Manag. Sci.* **36** (1990) 519–554.

[30] J. Robinson and A. Bernstein, A class of binary recurrent codes with limited error propagation. *IEEE Trans. Inf. Theory* **13** (1967) 106–113.

[31] J. Shearer, Some new optimum golomb rulers. *IEEE Trans. Inf. Theory* **36** (1990) 183–184.

[32] H.D. Sherali and W.P. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.* **3** (1990) 411–430.

[33] H.D. Sherali and W.P. Adams, A hierarchy of relaxations and convex hull characterizations for mixed-integer zero–one programming problems. *Discrete Appl. Math.* **52** (1994) 83–106.

[34] M.R. Slusky and W.-J. van Hoeve, A Lagrangian Relaxation for Golomb Rulers. (2013) 251–267.

[35] B.M. Smith, K. Stergiou and T. Walsh, Using auxiliary variables and implied constraints to model non-binary problems. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press (2000) 182–187.

[36] S.W. Soliday, A. Homaifar and G.L. Lebby, Genetic algorithm approach to the search for golomb rulers. In: *ICGA*. (1995) 528–535.