Astola and Tabus *SpringerPlus (2016)5:246*
DOI 10.1186/s40064-016-1863-8

SpringerPlus
a SpringerOpen Journal

RESEARCH Open Access

On the linear programming bound for linear Lee codes

Helena Astola* and Ioan Tabus

*Correspondence:
helena.astola@gmail.com
Department of Signal Processing, Tampere University of Technology, 33101 Tampere, Finland

Abstract

Based on an invariance-type property of the Lee-compositions of a *linear* Lee code, additional equality constraints can be introduced to the linear programming problem of linear Lee codes. In this paper, we formulate this property in terms of an action of the multiplicative group of the field $\mathbb{F}_q$ on the set of Lee-compositions. We show some useful properties of certain sums of Lee-numbers, which are the eigenvalues of the Lee association scheme, appearing in the linear programming problem of linear Lee codes. Using the additional equality constraints, we formulate the linear programming problem of linear Lee codes in a very compact form, leading to a fast execution, which allows to efficiently compute the bounds for large parameter values of the linear codes.

Keywords: Lee codes, Linear codes, Linear programming bound , Lee-compositions, Lee-numbers

Background

Finding the largest code (in cardinality) with a given length and minimum distance is one of the most fundamental problems in coding theory. The most well-known upper bounds for the Hamming metric are the Hamming bound, Plotkin bound, Singleton bound and Elias bound, and these bounds have been formulated for the Lee metric also, although the expressions are slightly more complicated. Delsarte (1973) introduced association schemes to coding theory to deal with topics involving the inner distribution of a code. An important approach arises from association schemes to the problem of determining the upper bound for the size of a code, namely the linear programming approach. The asymptotically best upper bound in the Hamming metric is the McEliece–Rodemich–Rumsey–Welch bound, see McEliece et al. (1977), which is based on the linear programming approach, and gives a substantial improvement to the earlier best upper bound, which is the Elias bound. In the Hamming metric, the distance relations between codewords directly define an association scheme, but in the Lee metric, the distance relations between the Lee-compositions define the association scheme. The Lee association scheme and linear programming bounds for Lee codes have been discussed by Astola (1982a), Solé (1988), and Tarnanen (1982). Generalizing to finite Frobenius rings, the linear programming bound for codes equipped with homogeneous weight, including the Lee weight on $\mathbb{Z}_4$, has been studied by Byrne et al. (2007).

We denote $\mathbb{Z}_q^n = \{\mathbf{x} = [x_1, \ldots, x_n] | x_i \in \{0, 1, \ldots, q-1\}\}$ the set of length n vectors having $q-$ary elements. The Hamming distance between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$ is $d_H(\mathbf{x}, \mathbf{y}) = |\{i \mid x_i \neq y_i\}|$, and penalizes equally any non-zero error between the components of $\mathbf{x}$ and $\mathbf{y}$, being a natural measure for errors in many communication channels. However, in communication channels with phase modulation, where the symbols $0, 1, \ldots, q-1$ are transmitted as phases $0, \frac{2\pi}{q}, \ldots, (q-1)\frac{2\pi}{q}$, the errors are measured as the shortest distance between the symbols along the unit circle and hence we consider in the following the error between two symbols $i, j \in \mathbb{Z}_q$ to be defined as the Lee distance $d_L(i, j) = \min(|i-j|, q-|i-j|)$, which was introduced by Lee (1958). The Lee distance between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$ is defined as

$$d_L(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \min\left(|x_i - y_i|, q - |x_i - y_i|\right).$$

The Lee weight of one symbol $j \in \mathbb{Z}_q$ is $w_L(j) = \min(j, q-j)$ and that of one vector $\mathbf{x} \in \mathbb{Z}_q^n$ is $w_L(\mathbf{x}) = d_L(\mathbf{0}, \mathbf{x})$. The number of components of the vector $\mathbf{x} \in \mathbb{Z}_q^n$ having the Lee weight equal to i is denoted $l_i(\mathbf{x})$ and, since $l_i(\mathbf{x}) = l_{q-i}(\mathbf{x})$, only $s = \lfloor \frac{q}{2} \rfloor$ such numbers describe completely the distribution of the Lee weights of the elements of $\mathbf{x}$. This distribution, denoted $l(\mathbf{x})$, is called the Lee-composition of $\mathbf{x}$, defined as:

$$l(\mathbf{x}) = [l_0(\mathbf{x}), l_1(\mathbf{x}), \ldots, l_s(\mathbf{x})].$$

We denote $\mathcal{L}_q^n = \{\mathbf{t}_0, \ldots, \mathbf{t}_\alpha\} = \{l(\mathbf{x}) | \mathbf{x} \in \mathbb{Z}_q^n\}$ the set of distinct Lee-compositions, where $(\alpha + 1)$ is their number, and we reserve the notation $\mathbf{t}_0$ for $l(\mathbf{0}) = [n, 0, \ldots, 0]$.

A code is defined as a subset of vectors of $\mathbb{Z}_q^n$. Keeping with the communication scenario, only the codewords belonging to the code C are sent as messages over the communication channel, say the codeword $\mathbf{x} \in C$ is sent and is corrupted due to communication errors, being received as a different vector, $\mathbf{z} \in \mathbb{Z}_q^n$. If the minimum distance d between all codewords of C is $2e+1$, then all the spheres $\{\mathbf{y} | d_L(\mathbf{x}, \mathbf{y}) \leq e, \mathbf{x} \in C\}$ around each codeword $\mathbf{x} \in C$ are non-intersecting, and the received vector $\mathbf{z}$ can be correctly decoded if $d_L(\mathbf{z}, \mathbf{x}) \leq e$. When designing codes C, the minimum distance d of the code is imposed as an initial requirement. The goal is then to design a code having the largest number of codewords, so that one can send reliably as many messages as possible, with the guarantee of correcting errors of Lee weight smaller than or equal to e. Since the problem of designing the largest codes is not solved yet in general, finding upper bounds on the size of the code C for a given d is considered one of the major problems in information theory. For example when new codes are proposed, comparing their size against the known upper bounds allows to prove their optimality if they achieve the upper bounds.

In the case of Lee distance, the constraints that need to be satisfied by an error correcting code with a given distance d can be conveniently analyzed using the association scheme introduced by Delsarte (1973). The scheme has $\alpha + 1$ relations denoted $K_{\mathbf{t}_0}, \ldots, K_{\mathbf{t}_\alpha}$, where $(\mathbf{x}, \mathbf{y})$ belongs to $K_{\mathbf{t}_i}$ if $l(\mathbf{x} - \mathbf{y}) = \mathbf{t}_i$, with $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$ and $\mathbf{t}_i \in \mathcal{L}_q^n$.

Introduce the inner distribution of the code C as

$$B_{\mathbf{t}_i} = \frac{1}{|C|}|K_{\mathbf{t}_i} \cap C^2|, \quad \mathbf{t}_i \in \mathcal{L}_q^n,$$

where $|C|$ denotes the cardinality of the set C, and in particular $B_{\mathbf{t}_0} = 1$. Hence the desired quantity to be optimized for a code, its cardinality $|C|$, can be simply expressed in terms of the variables $B_{\mathbf{t}_i}$ as

$$\sum_{\mathbf{t}_i \in \mathcal{L}_q^n} B_{\mathbf{t}_i} = |C|.$$

The fundamental result of the association scheme introduced by Delsarte (1973) is the formulation of the linear constraints between the variables $B_{\mathbf{t}_i}$, using as coefficients the so-called Lee-numbers, which are the eigenvalues of the Lee association scheme. Let $\xi = exp(2\pi\sqrt{-1}/q)$. The Lee-number $L_{\mathbf{t}}(\mathbf{u})$ with $\mathbf{t}, \mathbf{u} \in \mathcal{L}_q^n$ can be computed as follows: take any vector $\mathbf{v}$ having the Lee-composition $\mathbf{u} = l(\mathbf{v})$ and compute (see Astola 1982b)

$$L_{\mathbf{t}}(\mathbf{u}) = \sum_{\mathbf{x}|l(\mathbf{x})=\mathbf{t}} \left(\prod_{i=1}^{n} \xi^{v_i x_i} \right). \tag{1}$$

The linear inequality constraints between the variables $B_{\mathbf{t}_i}$ are then expressed as (see Astola 1982b):

$$\sum_{i=1}^{\alpha} B_{\mathbf{t}_i} L_{\mathbf{k}}(\mathbf{t}_i) \geq -\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix},$$

where $\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} = \binom{n}{\mathbf{k}} 2^{n-k_0}$ for $q = 2s+1$, and $\binom{n}{\mathbf{k}}$ is the multinomial coefficient $\binom{n}{k_0,\ldots,k_s}$.

With these constraints, the cardinality $\sum_{\mathbf{t}_i \in \mathcal{L}_q^n} B_{\mathbf{t}_i} = |C|$ of any code can be upper bounded by solving a linear programming problem, as stated in the following (see Astola 1982b). Let $B^*_{\mathbf{t}}$, $\mathbf{t} \in \{\mathbf{t}_1, \ldots, \mathbf{t}_\alpha\}$ be an optimal solution of the linear programming problem

$$\begin{aligned} &\text{maximize} \sum_{i=1}^{\alpha} B_{\mathbf{t}_i} \\ &B_{\mathbf{t}_i} \geq 0, \quad i \in I \text{ and } B_{\mathbf{t}_i} = 0, \quad i \in \{1, \ldots, \alpha\} \backslash I \\ &\sum_{i=1}^{\alpha} B_{\mathbf{t}_i} L_{\mathbf{k}}(\mathbf{t}_i) \geq -\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix}, \quad \text{for all } \mathbf{k} \in \mathcal{L}_q^n \end{aligned} \tag{2}$$

where $I = \{i \mid l(\mathbf{x}) = \mathbf{t}_i \Rightarrow w_L(\mathbf{x}) \geq d\}$. Then $1 + \sum_{i=1}^{\alpha} B^*_{\mathbf{t}_i}$ is an upper bound to the size of the code C with the minimum distance d.

We have previously shown that in the case of *linear* Lee codes we can use the following sharpening of the linear programming problem (see Astola and Tabus 2015). Let $B^*_{\mathbf{t}}$, $\mathbf{t} \in \{\mathbf{t}_1, \ldots, \mathbf{t}_\alpha\}$ be an optimal solution of the linear programming problem

$$\begin{aligned} &\text{maximize} \sum_{i=1}^{\alpha} B_{\mathbf{t}_i} \\ &B_{\mathbf{t}_i} \geq 0, \quad i \in I \text{ and } B_{\mathbf{t}_i} = 0, \quad i \in \{1, \ldots, \alpha\} \backslash I \end{aligned} \tag{3}$$

$$B_{\mathbf{t}_i} = B_{\mathbf{t}_j} \quad \text{for all } \mathbf{t}_j \in \tau(\mathbf{t}_i) \tag{4}$$

$$\sum_{i=1}^{\alpha} B_{\mathbf{t}_i} L_{\mathbf{k}}(\mathbf{t}_i) \geq -\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix}, \quad \text{for all } \mathbf{k} \in \mathcal{L}_q^n \tag{5}$$

where $\tau(\mathbf{t}_i)$ is the set of Lee-compositions that are obtained when the vectors $\mathbf{x}$ having the Lee-composition $\mathbf{t}_i$ are multiplied by all $r \in \mathbb{F}_q \backslash \{0\}$, and $I = \{i \mid l(\mathbf{x}) = \mathbf{t}_i \Rightarrow w_L(\mathbf{x}) \geq d\}$, i.e., the indices of those Lee-compositions corresponding to vectors of Lee weight $\geq d$. Then $1 + \sum_{i=1}^{\alpha} B_{\mathbf{t}_i}^*$ is an upper bound to the size of the code C with the minimum distance d.

The above sharpening is based on an invariance-type property of the Lee-compositions of a linear code. In the Hamming metric, multiplying codewords by a constant does not change the weight of the codewords. However, in the Lee metric, multiplication typically changes the Lee-composition of the codeword and so also usually the Lee weight. With linear codes, since they are linear subspaces of vector spaces, all the multiplied versions of any codeword also belong to the code. The authors have previously shown that there are as many codewords having the Lee-composition of a given codeword $\mathbf{x}$ as there are codewords having the Lee-composition of $r\mathbf{x}$ that is obtained by multiplication of the given codeword $\mathbf{x}$ by some constant r (see Astola and Tabus 2015).

In this paper, we formulate this property of Lee-compositions of a *linear* Lee code in terms of an action of the multiplicative group of the field $\mathbb{F}_q$ on the set of Lee-compositions. This formulation gives theoretical tools for studying the Lee-compositions and linear Lee-codes. For simplicity, we let q be prime, $\mathbb{F}_q^n = \{\mathbf{x} \mid x_i \in \mathbb{F}_q, i = 1, \ldots, n\}$. In addition, we show some useful properties of certain sums of Lee-numbers. Using the equality constraints introduced by Astola and Tabus (2015) and the properties of certain sums of Lee-numbers, we may compact the set of variables and linear constraints in the linear programming problem, and perform all computations with rational numbers. Compacting the problem leads to a faster execution, which allows to efficiently compute the bounds for large parameter values.

The group action

In the paper by Astola and Tabus (2015), a mapping $\tau(\mathbf{t})$ that maps the Lee-composition $\mathbf{t}$ into the set of Lee-compositions, which are obtained from $\mathbf{t}$ by multiplication of vectors having the Lee-composition $\mathbf{t}$ by all $r \in \{1, \ldots, q-1\}$, was defined in the following way:

$$\begin{aligned} \tau(\mathbf{t}) = &\big\{ \left[t_{\pi_r(0)}(\mathbf{x}), t_{\pi_r(1)}(\mathbf{x}), \ldots, t_{\pi_r(s)}(\mathbf{x}) \right] \mid \mathbf{t} = l(\mathbf{x}), \pi_r(i) = |k|, \\ &kr \equiv i \mod q, -s \leq k \leq s,\ 1 \leq r \leq q-1 \big\}, \end{aligned}$$

where

$$\pi_r(i) = |k| \text{ such that } rk \equiv i \mod q \text{ and } -s \leq k \leq s.$$

The mapping of the Lee-compositions can be formulated in terms of a group action.

Definition 1 If G is a group and X is a set, then a group action φ of G on X is a function

$$\varphi : G \times X \to X$$

that satisfies the following conditions for all $x \in X$:

1. $\varphi(e, x) = x$, where e is the identity element of G (identity).
2. $\varphi(g, \varphi(h, x)) = \varphi(gh, x)$ for all $g, h \in G$ (compatibility).

For notational reasons, denote in the following the set of Lee-compositions $\mathcal{L}_q^n$ as $\{\rho^{(0)}, \ldots, \rho^{(\alpha)}\}$. Now, let us define the function φ as follows. Denote by $\mathbb{F}_q^*$ the multiplicative group of the field $\mathbb{F}_q$.

$$\varphi : \mathbb{F}_q^* \times \left\{\rho^{(0)}, \ldots, \rho^{(\alpha)}\right\} \to \left\{\rho^{(0)}, \ldots, \rho^{(\alpha)}\right\},$$

where

$$\varphi(r, \rho^{(i)}) = \left[\rho_0^{(i)}, \rho_{\pi_{r(1)}}^{(i)}, \ldots, \rho_{\pi_{r(s)}}^{(i)}\right] = \rho^{(j)},$$

where

$$\pi_r(l) = |k| \text{ such that } rk \equiv l \mod q \text{ and } -s \le k \le s,\ r \in \mathbb{F}_q^*.$$

Lemma 1 *The function φ is a group action of $\mathbb{F}_q^*$, where q is prime, on the set $\{\rho^{(0)}, \ldots, \rho^{(\alpha)}\}$ of Lee-compositions.*

Let us show that the above function φ is in fact a group action. Clearly the identity property is satisfied as

$$\varphi(1, \rho^{(i)}) = \left[\rho_0^{(i)}, \rho_1^{(i)}, \ldots, \rho_s^{(i)}\right] = \rho^{(i)}.$$

The compatibility property requires that $\varphi(r_1, \varphi(r_2, \rho^{(i)})) = \varphi(r_1 \cdot r_2, \rho^{(i)})$, where $r_1, r_2 \in \mathbb{F}_q^*$. Let us write

$$\varphi(r_2, \rho^{(i)}) = \left[\rho_0^{(i)}, \rho_{\pi_{r_2(1)}}^{(i)}, \ldots, \rho_{\pi_{r_2(s)}}^{(i)}\right].$$

Then,

$$\varphi(r_1, (\varphi(r_2, \rho^{(i)})) = \left[\rho_0^{(i)}, \rho_{\pi_{r_1\left(\pi_{r_2(1)}\right)}}^{(i)}, \ldots, \rho_{\pi_{r_1\left(\pi_{r_2(s)}\right)}}^{(i)}\right].$$

Now,

$$\varphi(r_1 \cdot r_2, \rho^{(i)}) = \left[\rho_0^{(i)}, \rho_{\pi_{r_1 \cdot r_2(1)}}^{(i)}, \ldots, \rho_{\pi_{r_1 \cdot r_2(s)}}^{(i)}\right].$$

Therefore, we need to show that $\pi_{r_1(\pi_{r_2(l)})} = \pi_{r_1 \cdot r_2(l)}$.

Now,

$$\begin{cases} \pi_{r_2}(l) = |k_1| & \text{such that } r_2 k_1 \equiv l \mod q,\ -s \le k_1 \le s, \\ \pi_{r_1}(\pi_{r_2}(l)) = |k_2| & \text{such that } r_1 k_2 \equiv |k_1| \mod q,\ -s \le k_2 \le s, \\ \pi_{r_1 r_2}(l) = |k_3| & \text{such that } r_1 r_2 k_3 \equiv l \mod q,\ -s \le k_3 \le s. \end{cases}$$

We need to prove that $|k_2| = |k_3|$.

Proof We have two cases. If $1 \leq k_1 \leq s$ we have

$$r_1 k_2 \equiv k_1 \mod q \Rightarrow r_2 k_1 r_1 k_2 \equiv k_1 l \mod q \Rightarrow r_2 r_1 k_2 \equiv l \mod q,$$

and since q is prime and $-s \leq k_2, k_3 \leq s$ it must be that $k_2 = k_3$.

If $-s \leq k_1 < 0$ we have

$$r_1 k_2 \equiv -k_1 \mod q \Rightarrow r_2 k_1 r_1 k_2 \equiv -k_1 l \mod q \Rightarrow r_2 r_1 (-k_2) \equiv l \mod q,$$

and since q is prime and $-s \leq k_2, k_3 \leq s$ it must be that $|k_2| = |k_3|$. □

The action of $\mathbb{F}_q^*$ on the set of Lee-compositions partitions the set into equivalence classes, which are called orbits. So, the orbit of an element $\rho^{(i)}$ is

$$Orb(\rho^{(i)}) = \left\{ \varphi\left(r, \rho^{(i)}\right) : r \in \mathbb{F}_q^* \right\}.$$

Clearly there is a correspondence between $\tau(\mathbf{t}_i)$ and the orbits $Orb(\rho^{(i)})$, i.e., the sets $\tau(\mathbf{t})$ are the orbits of the above group action. Therefore, the theory of group actions can be used for studying the Lee-compositions and the linear programming problem, e.g., for the compact problem that we introduce in this paper, the orbit-counting theorem (see, for instance, Burnside 1897) can be used for determining the complexity of the problem as the number of orbits equals the number of variables in the compact problem.

Properties of certain sums of Lee-numbers

In this section we study certain sums of Lee-numbers, where the Lee-numbers are taken over Lee-compositions belonging to a given orbit. We show that this type of a sum is rational as opposed to the Lee-numbers, which in many cases are irrational numbers, and that it satisfies certain equality constraints.

Since the values of the coefficients of the inner distribution are equal for all compositions in $\tau(\mathbf{t}_i)$, we will have in (5) sums of the form

$$B_{\mathbf{t}_i} \cdot \left(L_{\mathbf{k}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}}(\mathbf{t}_{i_u}) \right).$$

Therefore, for each coefficient $B_{\mathbf{t}_i}$ we are computing a sum of Lee-numbers, where the Lee-numbers are taken over the Lee-compositions belonging to $\tau(\mathbf{t}_i)$.

Let us introduce the following lemma.

Lemma 2 *Let* $\tau(\mathbf{t}_i) = \{\mathbf{t}_{i_1}, \ldots, \mathbf{t}_{i_u}\}$. *Then the sum*

$$L_{\mathbf{k}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}}(\mathbf{t}_{i_u}), \tag{6}$$

is a rational number.

Proof First, we want to change each term $L_{\mathbf{k}}(\mathbf{t}_i)$ into $L_{\mathbf{t}_i}(\mathbf{k})$. There is the following relationship between the Lee-numbers $L_{\mathbf{k}}(\mathbf{t})$ and $L_{\mathbf{t}}(\mathbf{k})$ (see Astola 1982b):

$$\begin{bmatrix} n \\ \mathbf{t} \end{bmatrix} L_{\mathbf{k}}(\mathbf{t}) = \begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} L_{\mathbf{t}}(\mathbf{k}), \tag{7}$$

where the coefficient $\begin{bmatrix} n \\ \mathbf{t} \end{bmatrix}$ corresponds to the number of vectors in $\mathbb{F}_q^n$ for which the Lee-composition is $\mathbf{t}$ (see Astola 1982b). Hence, we can write (6) as

$$\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} / \begin{bmatrix} n \\ \mathbf{t}_{i_1} \end{bmatrix} L_{\mathbf{t}_{i_1}}(\mathbf{k}) + \begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} / \begin{bmatrix} n \\ \mathbf{t}_{i_2} \end{bmatrix} L_{\mathbf{t}_{i_2}}(\mathbf{k}) + \cdots + \begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} / \begin{bmatrix} n \\ \mathbf{t}_{i_u} \end{bmatrix} L_{\mathbf{t}_{i_u}}(\mathbf{k}).$$

Now, we notice that since all $\mathbf{t}_i$ belong to the same τ, they are permutations of each other. This means that the coefficients $\begin{bmatrix} n \\ \mathbf{t}_i \end{bmatrix}$ are all equal and (6) takes the form

$$\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} / \begin{bmatrix} n \\ \mathbf{t}_{i_1} \end{bmatrix} \left(L_{\mathbf{t}_{i_1}}(\mathbf{k}) + L_{\mathbf{t}_{i_2}}(\mathbf{k}) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}) \right),$$

where $\begin{bmatrix} n \\ \mathbf{k} \end{bmatrix} / \begin{bmatrix} n \\ \mathbf{t}_{i_1} \end{bmatrix} = 1$ if $\mathbf{k} \in \tau(\mathbf{t}_i)$, and a rational number otherwise.

Now, using Eq. (1), we can write the sum in parentheses above as

$$\begin{aligned} &L_{\mathbf{t}_{i_1}}(\mathbf{k}) + L_{\mathbf{t}_{i_2}}(\mathbf{k}) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}) \\ &\quad = \sum_{\mathbf{x} \mid l(\mathbf{x}) = \mathbf{t}_{i_1}} \left(\prod_{i=1}^{n} \xi^{v_i x_i} \right) + \sum_{\mathbf{x} \mid l(\mathbf{x}) = \mathbf{t}_{i_2}} \left(\prod_{i=1}^{n} \xi^{v_i x_i} \right) + \cdots + \sum_{\mathbf{x} \mid l(\mathbf{x}) = \mathbf{t}_{i_u}} \left(\prod_{i=1}^{n} \xi^{v_i x_i} \right) \end{aligned} \tag{8}$$

where u is the cardinality of $\tau(\mathbf{t}_i)$ and $l(\mathbf{v}) = \mathbf{k}$.

Since the Lee-numbers $L_{\mathbf{t}_i}(\mathbf{k})$ correspond to compositions according to τ, then for each vector $\mathbf{x}$, the sum in (8) includes all the vectors $r\mathbf{x}$, where $r \in \{1, \ldots, q-1\}$. Also, each vector can only have one Lee-composition and thus can appear in only one of the sums in (8).

We may now rearrange and group the terms in (8) as

$$\left(\xi^{\mathbf{v}\cdot\mathbf{x}_1} + \xi^{\mathbf{v}\cdot 2\mathbf{x}_1} + \cdots + \xi^{\mathbf{v}\cdot(q-1)\mathbf{x}_1} \right) + \left(\xi^{\mathbf{v}\cdot\mathbf{x}_2} + \xi^{\mathbf{v}\cdot 2\mathbf{x}_2} + \cdots + \xi^{\mathbf{v}\cdot(q-1)\mathbf{x}_2} \right) + \cdots$$

This forms a partition of the set of vectors having a Lee-composition in $\tau(\mathbf{t}_i)$, since the relation $\mathcal{R}$ defined as $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}_x$ iff $\mathbf{x} = r\mathbf{y}$, $r \in \{1, \ldots, q-1\}$ is clearly an equivalence relation.

Therefore, we can group the sum into m parts, each having $q-1$ terms. We now take one such part:

$$\xi^{\mathbf{v}\cdot\mathbf{x}_i} + \xi^{\mathbf{v}\cdot 2\mathbf{x}_i} + \cdots + \xi^{\mathbf{v}\cdot(q-1)\mathbf{x}_i},$$

and write it as

$$\xi^{\mathbf{v}\cdot\mathbf{x}_i} + \xi^{2(\mathbf{v}\cdot\mathbf{x}_i)} + \cdots + \xi^{(q-1)(\mathbf{v}\cdot\mathbf{x}_i)}.$$

If $\mathbf{v} \cdot \mathbf{x}_i = 0$ it equals $q-1$, otherwise, we have a sum of the form

$$\xi + \xi^2 + \cdots + \xi^{q-1} = -1.$$

So (8) is a sum of the form $m_1(q-1) + m_2(-1)$, where m_1 and m_2 are integers ≥ 0 such that $m = m_1 + m_2$. □

Furthermore, we notice that when we look at the sums

$$\begin{aligned} &L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_u}), \\ &L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_u}), \\ &\qquad\qquad \vdots \\ &L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_u}), \end{aligned} \tag{9}$$

where u is the cardinality of $\tau(\mathbf{t}_i)$ and u' is the cardinality of $\tau(\mathbf{k}_i)$, they all turn out be equal.

Lemma 3 *For* $\mathbf{t}_{i_1}, \ldots, \mathbf{t}_{i_u} \in \tau(\mathbf{t}_i)$ *and* $\mathbf{k}_i, \mathbf{k}_{i'} \in \tau(\mathbf{k}_i)$,

$$L_{\mathbf{k}_i}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_i}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_i}(\mathbf{t}_{i_u}) = L_{\mathbf{k}_{i'}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i'}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i'}}(\mathbf{t}_{i_u}).$$

Proof First, we transform these sums using (7) into sums of the following form

$$\begin{bmatrix} n \\ \mathbf{k}_i \end{bmatrix} \Big/ \begin{bmatrix} n \\ \mathbf{t}_{i_1} \end{bmatrix} \Big(L_{\mathbf{t}_{i_1}}(\mathbf{k}_i) + L_{\mathbf{t}_{i_2}}(\mathbf{k}_i) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}_i) \Big),$$
$$\begin{bmatrix} n \\ \mathbf{k}_{i'} \end{bmatrix} \Big/ \begin{bmatrix} n \\ \mathbf{t}_{i_1} \end{bmatrix} \Big(L_{\mathbf{t}_{i_1}}(\mathbf{k}_{i'}) + L_{\mathbf{t}_{i_2}}(\mathbf{k}_{i'}) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}_{i'}) \Big).$$

Now we notice that since the Lee-compositions $\mathbf{k}_i$ and $\mathbf{k}_{i'}$ both belong to $\tau(\mathbf{k}_i)$, the coefficients in front of the sums are all equal. What remains to show is that the sums $(L_{\mathbf{t}_{i_1}}(\mathbf{k}_i) + L_{\mathbf{t}_{i_2}}(\mathbf{k}_i) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}_i))$ and $(L_{\mathbf{t}_{i_1}}(\mathbf{k}_{i'}) + L_{\mathbf{t}_{i_2}}(\mathbf{k}_{i'}) + \cdots + L_{\mathbf{t}_{i_u}}(\mathbf{k}_{i'}))$ are equal. To show this, we rearrange both sums as we did in the previous proof to obtain sums of the following form. For the first sum we have

$$\left(\xi^{\mathbf{v}\cdot\mathbf{x}_1} + \xi^{2(\mathbf{v}\cdot\mathbf{x}_1)} + \cdots + \xi^{(q-1)\mathbf{v}\cdot\mathbf{x}_1}\right) + \left(\xi^{\mathbf{v}\cdot\mathbf{x}_2} + \xi^{2(\mathbf{v}\cdot\mathbf{x}_2)} + \cdots + \xi^{(q-1)\mathbf{v}\cdot\mathbf{x}_2}\right) + \cdots,$$

where $l(\mathbf{v}) = \mathbf{k}_i$ and $l(\mathbf{x}_i) = \mathbf{t}_i$.

Now, since the Lee-compositions $\mathbf{k}_i$ and $\mathbf{k}_{i'}$ belong to $\tau(\mathbf{k}_i)$, we obtain the vectors having the Lee-composition $\mathbf{k}_{i'}$ from the vectors having the Lee-composition $\mathbf{k}_i$ by multiplication by some r. Therefore, for the second sum we have

$$\left(\xi^{r\mathbf{v}\cdot\mathbf{x}_1} + \xi^{r2(\mathbf{v}\cdot\mathbf{x}_1)} + \cdots + \xi^{r(q-1)\mathbf{v}\cdot\mathbf{x}_1}\right) + \left(\xi^{r\mathbf{v}\cdot\mathbf{x}_2} + \xi^{r2(\mathbf{v}\cdot\mathbf{x}_2)} + \cdots + \xi^{r(q-1)\mathbf{v}\cdot\mathbf{x}_2}\right) + \cdots.$$

Now

$$\xi^{\mathbf{v}\cdot\mathbf{x}_i} + \xi^{2(\mathbf{v}\cdot\mathbf{x}_i)} + \cdots + \xi^{(q-1)\mathbf{v}\cdot\mathbf{x}_i} = \xi^{r\mathbf{v}\cdot\mathbf{x}_i} + \xi^{r2(\mathbf{v}\cdot\mathbf{x}_i)} + \cdots + \xi^{r(q-1)\mathbf{v}\cdot\mathbf{x}_i},$$

since if $\mathbf{v}\cdot\mathbf{x}_i = 0$, they are both equal to $q-1$, and otherwise each exponent is distinct. Therefore, the sums are equal. □

The compact linear programming problem

The additional equalities for linear Lee codes in (4) can be used for compacting the set of linear constraints in the linear programming problem of linear Lee codes. We enforce (3) by replacing all variables $B_{\mathbf{t}_j}$ for $\mathbf{t}_j \in \tau(\mathbf{t}_i)$ with a single variable γ_i, so that $\gamma_i = B_{\mathbf{t}_j}$ for all $\mathbf{t}_j \in \tau(\mathbf{t}_i)$.

We notice that by replacing the α variables $B_{\mathbf{t}_1}, \ldots, B_{\mathbf{t}_\alpha}$ of the linear programming problem with the set of variables $\gamma_1, \ldots, \gamma_\kappa$, where κ is the number of orbits, i.e., the number of different sets τ_i, we are eliminating the equality constraints from the sharpened linear programming problem. Let us denote by Υ a matrix, where we have listed all the Lee-numbers as

$$\Upsilon = \begin{bmatrix} L_{\mathbf{k_0}}(\mathbf{t_0}) & L_{\mathbf{k_0}}(\mathbf{t_1}) & \cdots & L_{\mathbf{k_0}}(\mathbf{t}_\alpha) \\ L_{\mathbf{k_1}}(\mathbf{t_0}) & L_{\mathbf{k_1}}(\mathbf{t_1}) & \cdots & L_{\mathbf{k_1}}(\mathbf{t}_\alpha) \\ \vdots & & \ddots & \vdots \\ L_{\mathbf{k}_\alpha}(\mathbf{t_0}) & L_{\mathbf{k}_\alpha}(\mathbf{t_1}) & \cdots & L_{\mathbf{k}_\alpha}(\mathbf{t}_\alpha) \end{bmatrix}$$

Let $\mathbf{A}$ be an $\alpha \times \kappa$ matrix of the form

$$\mathbf{A} = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \hline \cdots & & & & & \\ \hline 0 & 0 & 0 & 0 & \cdots & 1 \end{array}\right],$$

where the number of rows inside each partition corresponds to the cardinalities of the sets τ_i.

We introduce the vector $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_\kappa)$ and formulate the linear programming problem equivalent to (3):

$$\begin{aligned} &\text{maximize} \sum_{i=1}^{\kappa} |\tau_i| \gamma_i \\ &\gamma_0 = 1 \text{ and } \gamma_i \geq 0, \quad i \in I \text{ and } \gamma_i = 0, \quad i \in \{1, \ldots, \alpha\} \backslash I \\ &\Upsilon \mathbf{A} \gamma \geq 0, \end{aligned}$$

where $I = \{i \mid \forall \mathbf{t} \in \tau_i : l(\mathbf{x}) = \mathbf{t} \Rightarrow w_L(\mathbf{x}) \geq d\}$, i.e., the indices of those sets τ_i, where all Lee-compositions correspond to vectors of Lee weight $\geq d$.

The cardinalities $|\tau_i|$ appear in the criterion of the problem since the initial criterion expressed in terms of the variables $B_{\mathbf{t}_1}, \ldots, B_{\mathbf{t}_\alpha}$ is $\mathbf{1}^T\mathbf{B}$, where $\mathbf{1}$ is the all one vector and $\mathbf{B} = [B_{\mathbf{t}_1}, \ldots, B_{\mathbf{t}_\alpha}]^T$, and the criterion in the new variables is $\mathbf{1}^T\mathbf{A}\boldsymbol{\gamma}$, where the new vector of coefficients, $\mathbf{1}^T\mathbf{A}$, will have as elements the size of the partitions of $\mathbf{A}$, which are equal to the cardinalities of the sets τ_i.

Additionally we notice that the matrix $\mathcal{U} = \Upsilon\mathbf{A}$ can be seen to have the partition structure similar to that of A,

$$\mathcal{U} = \Upsilon\mathbf{A} = \Upsilon \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \hline \cdots & & & & & \\ \hline 0 & 0 & 0 & 0 & \cdots & 1 \end{array}\right] = \left[\begin{array}{cccccc} \Phi_{1,1} & \Phi_{1,2} & \Phi_{1,3} & \Phi_{1,4} & \cdots & \Phi_{1,\kappa} \\ \hline \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} & \Phi_{2,4} & \cdots & \Phi_{2,\kappa} \\ \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} & \Phi_{2,4} & \cdots & \Phi_{2,\kappa} \\ \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} & \Phi_{2,4} & \cdots & \Phi_{2,\kappa} \\ \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} & \Phi_{2,4} & \cdots & \Phi_{2,\kappa} \\ \hline \Phi_{3,1} & \Phi_{3,2} & \Phi_{3,3} & \Phi_{3,4} & \cdots & \Phi_{3,\kappa} \\ \Phi_{3,1} & \Phi_{3,2} & \Phi_{3,3} & \Phi_{3,4} & \cdots & \Phi_{3,\kappa} \\ \hline \cdots & & & & & \\ \hline \Phi_{\kappa,1} & \Phi_{\kappa,2} & \Phi_{\kappa,3} & \Phi_{\kappa,4} & \cdots & \Phi_{\kappa,\kappa} \end{array}\right]$$

$$= \mathbf{A}\Phi$$

where the matrix $\Phi = [\Phi]_{i,j}$ is $\kappa \times \kappa$ and the rows in a partition correspond to the Lee-compositions belonging to the same set τ_i. Inside a partition the rows of the matrix $\mathcal{U}$ are identical, due to Lemma 3, since the columns of $\mathcal{U}$ inside a partition have the following elements

$$\begin{aligned} &L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_1}}(\mathbf{t}_{i_u}),\\ &L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_2}}(\mathbf{t}_{i_u}),\\ &\qquad\qquad\vdots\\ &L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_1}) + L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_2}) + \cdots + L_{\mathbf{k}_{i_{u'}}}(\mathbf{t}_{i_u}), \end{aligned}$$

where $|\tau_{\mathbf{k}_i}| = u'$ and $|\tau_{\mathbf{t}_i}| = u$. This leads to a repeated inequality constraint. In order to remove this redundancy, we select from the matrix $\mathcal{U}$ only one row per partition block, keeping thus only the non-redundant inequalities. This is just the matrix Φ, and we may replace the inequality constraints $\Upsilon\mathbf{A}\boldsymbol{\gamma} \geq 0$ with $\Phi\boldsymbol{\gamma} \geq 0$. In addition, due to Lemma 2, each element of Φ is a rational number, and the Lee-numbers can be computed recursively with integers using the recursive formula given by Astola and Tabus (2013). Hence, we can perform all computations using integers instead of irrationals.

In Tables 1, 2, 3 and 4 there are numerical results for the linear programming bounds for linear Lee codes with $q = 5, 7, 13, 17$. The bounds were computed using the above compact linear programming bound for linear Lee codes and compared to the general linear programming bound given by (2). Italics indicates an improvement and * indicates a tight bound.

Table 1 Upper bounds for the dimension *k* of linear Lee codes when $q = 5$

n\d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Time [s]
2	1*														0.007
3	1*	1*													0.013
4	2*	2*	1*	1*											0.026
5	3*	3*	2*	1*	1*										0.041
6	4*	3*	3*	2*	*1**	1*	1*								0.068
7	5*	4*	3*	3*	2*	*1**	1*	1*							0.102
8	6*	5*	4*	4*	3*	*2**	2*	1*	1*	1*					0.169
9	7*	6*	5*	5	4*	3*	3	2*	*1**	1*	1*				0.249
10	8*	7*	6*	6	5	4	*3**	3	2*	2*	1*	1*	1*		0.408
11	9*	8*	7	*6**	6	5	4	4	3	*2**	2*	1*	1*	1*	0.563
12	10*	9*	8	7	7	6	5	5	4	3*	*2**	2*	2*	1*	
13	10*	10*	9	8	7	7	6	5	5	4	*3**	*2**	2*	*1**	
14	11*	11	10	9	8	8	7	6	*5*	5	4	*3**	3	2*	
15	12*	12	11	10	9	9	8	7	6	6	5	4	4	3	
n\d	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Time [s]**
12	1*	1*													0.872
13	1*	1*	1*												1.426
14	*1**	1*	1*	1*	1*										2.517
15	*2**	2*	1*	1*	1*	1*									3.951

The * indicates a tight bound and italics an improvement compared to the bound given by (2)

Table 2 Upper bounds for the dimension *k* of linear Lee codes when $q = 7$

n\d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Time [s]
2	1*														0.009
3	2*	1*	1*	1*											0.028
4	2*	2*	2*	1*	1*										0.047
5	3*	3*	2*	2*	*1**	1*	1*								0.098
6	4*	4*	3*	3*	2*	2*	*1**	1*	1*	1*					0.212
7	5*	5*	4*	4	3*	3	2*	2*	*1**	1*	1*				0.420
8	6*	*5**	5*	*4**	4*	4	3*	3	2*	2*	*1**	1*	1*		0.816
9	7*	*6**	6	5*	5	4*	4	3*	3	3	2*	*1**	*1**	1*	
10	8*	*7**	7	6*	6	5	5	4	4	3*	3	*2**	2*	*1**	
11	9*	*8**	8	7	*6**	*5**	5	5	5	4	4	3	3	*2**	
12	10*	9*	*8**	8	7	7	6	6	5	5	4	4	3	3	
13	11*	10*	9*	9	8	8	7	7	6	6	5	5	4	4	
14	12*	11*	10*	10	9	9	8	8	7	7	6	6	5	5	
15	13*	12*	11*	11	10	10	9	9	8	*7**	7	6	6	*5**	
n\d	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Time [s]**
9	1*	1*													1.621
10	1*	1*	1*												2.959
11	2*	2	1*	1*	1*										6.406
12	3	*2**	2	1*	1*	1*	1*	1*							17.580
13	*3**	3	*2**	*2**	2	1*	1*	1*	1*						31.983
14	4	4	3	3	*2**	2*	2	*1**	1*	1*	1*				67.885
15	5	5	4	4	3	3	*2**	*2**	*1**	*1**	1*	1*	1*	1*	173.448

The * indicates a tight bound and italics an improvement compared to the bound given by (2)

Table 3 Upper bounds for the dimension *k* of linear Lee codes when $q = 13$

n\d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Time [s]
2	1*	1*	1*												0.038
3	2*	*1**	1*	1*	1*	1*	1								0.107
4	3*	2*	2*	2*	2*	1*	1*	1*	1*	1*	1				0.487
5	4*	3*	3*	3	2*	2*	2*	2	1*	1*	1*	1*	1*	1	
6	5*	4*	4*	*3**	3*	3*	3	2*	2*	2*	2	*1**	1*	1*	
7	5*	5*	5	4*	4*	4	3*	3*	3*	3	2*	2*	2*	2*	
8	6*	6*	6	5*	5	5	4*	4	4	3*	3*	3	3	*2**	
n\d	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Time [s]**
5	1														2.288
6	1*	1*	1*	1*	1*										11.508
7	2	*1**	1*	1*	1*	1*									57.325
8	2*	2*	2*	2	2	1*	1*	1*	1*	1*					300.543

The * indicates a tight bound and italics an improvement compared to the bound given by (2)

In Tables 1, 2, 3 and 4, the computation time is also shown for each n. When computing the sharpened linear programming bound by giving additional equality constraints to the linear programming solver, more computation time is required for given parameters than with the general linear programming problem given by (2), since the

Table 4 Upper bounds for the dimension *k* of linear Lee codes when $q = 17$

n\d	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Time [s]
2	1*	1*	1*												0.043
3	2*	2*	1*	1*	1*	1*	1*								0.276
4	3*	*2**	2*	2*	2*	2*	1*	1*	1*	1*	1*	1*	1*		2.271
5	4*	3*	3*	3*	*2**	2*	2*	2*	2*	2	1*	1*	1*	1*	
6	5*	4*	4*	4*	3*	3*	3*	3	2*	2*	2*	2*	2*	2*	
n\d	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Time [s]**
5	1*	1*	1*												19.117
6	*1**	1*	1*	1*	1*	1*	1*	1*							188.785

The $*$ indicates a tight bound and italics an improvement compared to the bound given by (2)

linear programming solver is given a larger set of overall constraints. However, using the compact problem introduced in this paper reduces the computation time significantly, since there are less variables and constraints and all computations can be performed with rational numbers. For example, on an Apple iMac 5K (late 2014) with Intel Core i7 (I7-4790K), 32 gigabytes of memory, and a 64-bit operating system OS X Yosemite, and using the `linprog` linear programming solver of Matlab, for $q = 17$ and $n = 5$, computing the regular linear programming bounds took approximately 17 min and computing the sharpened bounds by using additional equality constraints took over 5 h. When using the compact problem, the computation time was less than 20 s, including computation of the new sets of constraints from the Lee-numbers according to the sets $\tau(\mathbf{t})$, which is a significant improvement and makes it possible to compute the bounds for larger parameter values in a reasonable time. The Lee-numbers were computed separately, and computing them recursively for $q = 17$ and $n = 2, \ldots, 6$ took approximately 1 min.

The obtained bounds can be used for identifying optimal codes. Consider the bound for k given in Table 4 with $q = 17$, $n = 5$ and $d = 7$, which is 2. This means that the maximum number of codewords in a linear code with these parameters is at most 289. The code having the generator matrix

$$\mathbf{G} = \left[\begin{array}{cc|ccc} 1 & 0 & 5 & 0 & 4 \\ 0 & 1 & 16 & 15 & 10 \end{array}\right]$$

is a [5, 2]-code with the minimum distance 7 and has $17^2 = 289$ codewords, therefore, it is an optimal linear code for the above parameters.

Conclusions

In this paper, we formulated the invariance-type property of Lee-compositions introduced by Astola and Tabus (2015) in terms of an action of the multiplicative group of the field $\mathbb{F}_q$ on the set of Lee-compositions. This formulation is useful in the theoretical study of Lee-compositions and linear Lee codes. In addition, we have shown some useful properties of certain sums of Lee-numbers that appear in the constraints of the linear programming problem of linear Lee codes. Based on the equality constraints and these properties, we constructed a more compact linear programming problem for linear Lee codes, leading to a fast execution and allowing all computations to be performed using

integers. This leads in practice to having available upper bounds for codes with parameters higher than available up to now.

Authors' contributions
Both authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 29 October 2015 Accepted: 16 February 2016
Published online: 01 March 2016

References

Astola H, Tabus I (2013) Bounds on the size of Lee-codes. In: 8thInternational symposium on image and signal processing and analysis,Trieste, Italy, pp 464–469
Astola H, Tabus I (2015) Sharpening the linear programming bound for linear Lee-codes. Electron Lett 51(6):492–494. doi:10.1049/el.2014.4300
Astola J (1982a) The Lee-scheme and bounds for Lee-codes. Cybern Syst 13(4):331–343. doi:10.1080/01969728208927711
Astola J (1982b) The theory of Lee-codes. Lappeenranta University of Technology, Department of Physics and Mathematics, Research Report 1/1982
Burnside W (1897) Theory of groups of finite order. Cambridge University Press, Cambridge
Byrne E, Greferath M, O'Sullivan ME (2007) The linear programming bound for codes over finite Frobenius rings. Des Codes Cryptogr 42(3):289–301
Delsarte P (1973) An algebraic approach to the association schemes of coding theory. Philips research reports: Supplements 10
Lee CY (1958) Some properties of nonbinary error-correcting codes. IRE Trans Inf Theory 4(2):77–82
McEliece R, Rodemich E, Rumsey H, Welch L (1977) New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities. IEEE Trans Inf Theory 23(2):157–166
Solé P (1988) The Lee association scheme. In: Cohen G, Godlewski P (eds) Coding theory and applications. Lecture notes in Computer science, vol 311. Springer, Berlin, pp 45–55. doi:10.1007/3-540-19368-5_5
Tarnanen H (1982) An approach to constant weight and Lee codes by using the methods of association schemes. Publications of University of Turku: Annales universitatis turkuensis. Series A I: Astronomica-chemica-physica-mathematica. University of Turku, Turku