On the Classification of Hermitian Self-Dual Additive Codes over GF(9)

Lars Eirik Danielsen

***Abstract*—Additive codes over $\mathbf{GF(9)}$ that are self-dual with respect to the Hermitian trace inner product have a natural application in quantum information theory, where they correspond to ternary quantum error-correcting codes. However, these codes have so far received far less interest from coding theorists than self-dual additive codes over $\mathbf{GF(4)}$, which correspond to binary quantum codes. Self-dual additive codes over $\mathbf{GF(9)}$ have been classified up to length 8, and in this paper we extend the complete classification to codes of length 9 and 10. The classification is obtained by using a new algorithm that combines two graph representations of self-dual additive codes. The search space is first reduced by the fact that every code can be mapped to a weighted graph, and a different graph is then introduced that transforms the problem of code equivalence into a problem of graph isomorphism. By an extension technique, we are able to classify all optimal codes of length 11 and 12. There are 56 005 876 $\mathbf{(11, 3^{11}, 5)}$ codes and 6493 $\mathbf{(12, 3^{12}, 6)}$ codes. We also find the smallest codes with trivial automorphism group.**

***Index Terms*—Self-dual codes, additive codes, codes over $\mathbf{GF(9)}$, graph theory, classification, nonbinary quantum codes.**

I. Introduction

Additive codes over $\mathrm{GF}(9)$ of length n are $\mathrm{GF}(3)$-linear subgroups of $\mathrm{GF}(9)^n$. Such an additive code contains 3^k codewords for some $0 \le k \le 2n$, and is called an $(n, 3^k)$ code. A code $\mathcal{C}$ can be defined by a $k \times n$ *generator matrix* with entries from $\mathrm{GF}(9)$ whose rows span $\mathcal{C}$ additively. We denote $\mathrm{GF}(9) = \{0, 1, \omega, \omega^2, \ldots, \omega^7\}$, where $\omega^2 = \omega + 1$. *Conjugation* of $x \in \mathrm{GF}(9)$ is defined by $\overline{x} = x^3$. The *trace map*, $\mathrm{Tr} : \mathrm{GF}(9) \to \mathrm{GF}(3)$, is defined by $\mathrm{Tr}(x) = x + \overline{x}$. Following Nebe, Rains, and Sloane [1], we define the *Hermitian trace inner product* of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathrm{GF}(9)^n$ by

$$(\boldsymbol{u}, \boldsymbol{v}) = \omega^2(\boldsymbol{u} \cdot \overline{\boldsymbol{v}} - \overline{\boldsymbol{u}} \cdot \boldsymbol{v}) = \mathrm{Tr}(\omega^2 \boldsymbol{u} \cdot \overline{\boldsymbol{v}}),$$

where multiplication by ω^2 is necessary because the skew-symmetric bilinear form $(\boldsymbol{u} \cdot \overline{\boldsymbol{v}} - \overline{\boldsymbol{u}} \cdot \boldsymbol{v})$ does not take values in $\mathrm{GF}(3)$ [1]. We define the *dual* of the code $\mathcal{C}$ with respect to the Hermitian trace inner product, $\mathcal{C}^\perp = \{\boldsymbol{u} \in \mathrm{GF}(9)^n \mid (\boldsymbol{u}, \boldsymbol{c}) = 0 \text{ for all } \boldsymbol{c} \in \mathcal{C}\}$. $\mathcal{C}$ is *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$. If $\mathcal{C} = \mathcal{C}^\perp$, then $\mathcal{C}$ is *self-dual* and must be an $(n, 3^n)$ code. The class of trace-Hermitian self-dual additive codes over $\mathrm{GF}(9)$ is also known as 9^{H+} [1]. The *Hamming weight* of $\boldsymbol{u}$, denoted $\mathrm{wt}(\boldsymbol{u})$, is the number of non-zero components of $\boldsymbol{u}$. The *Hamming distance* between $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\mathrm{wt}(\boldsymbol{u} - \boldsymbol{v})$. The *minimum distance* of the code $\mathcal{C}$ is the minimal Hamming distance between any two distinct codewords of $\mathcal{C}$. Since $\mathcal{C}$ is an additive code, the minimum distance is also given by the smallest non-zero weight of any codeword in $\mathcal{C}$. A code with minimum distance d is called an $(n, 3^k, d)$ code. The *weight distribution* of the code $\mathcal{C}$ is the sequence $(A_0, A_1, \ldots, A_n)$, where A_i is the number of codewords of weight i. The *weight enumerator* of $\mathcal{C}$ is the polynomial

$$W(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i,$$

where we will denote $W(y) = W(1, y)$. It follows from the *Singleton bound* [2] that any self-dual additive code must satisfy $d \le \lfloor \frac{n}{2} \rfloor + 1$. A code is called *extremal* if it has minimum distance $\lfloor \frac{n}{2} \rfloor + 1$, and *near-extremal* if it has minimum distance $\lfloor \frac{n}{2} \rfloor$. If a code has the highest possible minimum distance for the given length, it is called *optimal*. A tighter bound exists for codes over $\mathrm{GF}(4)$ [3], but in general the Singleton bound is the best known upper bound. Codes that satisfy the Singleton bound with equality are also known as *maximum distance separable (MDS) codes*. The well-known *MDS conjecture* implies that self-dual additive MDS codes over $\mathrm{GF}(9)$ must have length $n \le 10$. We have shown in previous work [4] that there are only three non-trivial MDS codes, with parameters $(4, 3^4, 3)$, $(6, 3^6, 4)$, and $(10, 3^{10}, 6)$, given that the MDS conjecture holds.

Two self-dual additive codes over $\mathrm{GF}(9)$ are *equivalent* if the codewords of one can be mapped onto the codewords of the other by a transformation that preserves the properties of the code, i.e., weight enumerator, additivity, and self-duality. It was shown by Rains [2] that this group of transformations is $\mathrm{Sp}_2(3) \wr \mathrm{Sym}(n)$, i.e., permutations of the coordinates combined with operations from the *symplectic group* $\mathrm{Sp}_2(3)$ applied independently to each coordinate. Global conjugation of all coordinates will also preserve the properties of the code, and codes related by this operation are called *weakly equivalent* [1]. In this paper, we classify codes up to equivalence, i.e., we do not consider global conjugation. Let an element $a + b\omega \in \mathrm{GF}(9)$, be represented as $\binom{a}{b} \in \mathrm{GF}(3)^2$. We can then premultiply this element by a 2×2 matrix. The group

$$\mathrm{Sp}_2(3) = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right\rangle$$

has order 24 and contains all 2×2 matrices with elements from $\mathrm{GF}(3)$ and determinant one. The order of $\mathrm{Sp}_2(3) \wr \mathrm{Sym}(n)$ is $24^n n!$, and hence this is the total number of maps that take a self-dual additive code over $\mathrm{GF}(9)$ to an equivalent code [2]. By translating the action of $\mathrm{Sp}_2(3)$ on $\binom{a}{b}$ into operations on elements $c = a + b\omega \in \mathrm{GF}(9)$, we find that the operations we can apply to all elements in a coordinate of a code are

In *IEEE Trans. Inform. Theory* 58(8), pp. 5500–5511, 2012.

Manuscript created June 13, 2011; revised November 29, 2011. This research was supported by the Research Council of Norway. The material in this paper was presented in part at the 2010 IEEE International Symposium on Information Theory, Austin, TX, USA, June 2010.

L.E. Danielsen is with the Department of Informatics, University of Bergen, N-5020 Bergen, Norway (e-mail: larsed@ii.uib.no).

$c \mapsto xc$ if $x^4 = 1$, or $c \mapsto x\overline{c}$ if $x^4 = -1$, given $x \in \mathrm{GF}(9)$, and $a + b\omega \mapsto a + yb + b\omega$, given $y \in \mathrm{GF}(3)$.

A transformation that maps $\mathcal{C}$ to itself is called an *automorphism* of $\mathcal{C}$. All automorphisms of $\mathcal{C}$ make up the *automorphism group* of $\mathcal{C}$, denoted $\mathrm{Aut}(\mathcal{C})$. The number of distinct codes equivalent to $\mathcal{C}$ is then given by $\frac{24^n n!}{|\mathrm{Aut}(\mathcal{C})|}$. The *equivalence class* of $\mathcal{C}$ contains all distinct codes that are equivalent to $\mathcal{C}$. By adding the sizes of all equivalence classes of codes of length n, we find the total number of distinct codes of length n, denoted T_n. The number T_n is also given by a *mass formula* which was described by Höhn [5] for self-dual additive codes over $\mathrm{GF}(4)$ and is easily generalized to $\mathrm{GF}(9)$:

$$T_n = \prod_{i=1}^{n} (3^i + 1) = \sum_{j=1}^{t_n} \frac{24^n n!}{|\mathrm{Aut}(\mathcal{C}_j)|}, \tag{1}$$

where t_n is the number of equivalence classes of codes of length n, and $\mathcal{C}_j$ is a representative from each equivalence class. The smallest possible automorphism group, called the *trivial automorphism group*, of a self-dual additive code over $\mathrm{GF}(9)$ is $\{I, -I\}$, i.e., it consists of global multiplication of coordinates by 1 or -1. By assuming that all codes of length n have a trivial automorphism group, we obtain from the mass formula a lower bound on t_n, the total number of inequivalent codes.

$$t_n \geq \left\lceil \frac{2 \prod_{i=1}^{n} (3^i + 1)}{24^n n!} \right\rceil. \tag{2}$$

Note that when n is large, most codes have a trivial automorphism group, so the tightness of the bound increases with n. As we will see in Section VII, for $n = 10$, 80% of all codes have a trivial automorphism group, and the bound (2) underestimates t_{10} by just 19%.

Any *linear* code over $\mathrm{GF}(9)$ that is self-dual with respect to the *Hermitian inner product*, $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \cdot \overline{\boldsymbol{v}}$, is also a self-dual additive code with respect to the Hermitian trace inner product. The class of Hermitian self-dual linear codes over $\mathrm{GF}(9)$ is also known as 9^H [1]. The operations that map a self-dual linear code to an equivalent code are more restrictive than for additive codes, since $\mathrm{GF}(9)$-linearity must now be preserved. Only coordinate permutations and multiplication of single coordinates by $x \in \mathrm{GF}(9)$ where $x^4 = 1$ is allowed. It follows that only additive codes that satisfy certain constraints can be equivalent to linear codes. Such constraints for codes over $\mathrm{GF}(4)$ were described by Van den Nest [6] and by Glynn et al. [7]. An obvious constraint is that all coefficients of the weight enumerator, except A_0, of a linear code must be divisible by 8, whereas for an additive code they need only be divisible by 2. To our knowledge, no complete classification of Hermitian self-dual linear codes over $\mathrm{GF}(9)$ have appeared so far, but several authors have studied this class of codes and suggested a number of constructions [8]–[11]. Checking whether a self-dual additive code over $\mathrm{GF}(9)$ is equivalent to a linear code is non-trivial, since there are 6^n coordinate transformations in $\mathrm{Sp}_2(3)^n$ that could transform a non-linear code into a linear code. Our classification of self-dual additive codes could be a useful starting point for also studying linear codes, but this is left as a problem for future work.

Trace-Hermitian self-dual additive codes over $\mathrm{GF}(q)$ exist for $q = m^2$, where m is a prime power [1], and the class of self-dual additive codes over $\mathrm{GF}(q)$ is called q^{H+}. The first case, 4^{H+}, has been studied in detail, in particular since an application to *quantum error-correction* was discovered [3]. We have previously classified self-dual additive codes over $\mathrm{GF}(4)$ up to length 12 [12]. Self-dual linear codes over $\mathrm{GF}(4)$ have been classified up to length 16 [13] by Conway, Pless, and Sloane. This classification was recently extended to length 18 [14] and 20 [15] by Harada et al. The next class of self-dual additive codes, 9^{H+}, has received less attention, although these codes have similar application in quantum error-correction [2], [16], where they correspond to *ternary quantum codes*. We have previously classified self-dual additive codes over $\mathrm{GF}(9)$ up to length 8 [4], as well as self-dual additive codes over $\mathrm{GF}(16)$ and $\mathrm{GF}(25)$ up to length 6. Another type of self-dual code over $\mathrm{GF}(9)$ is known as 9^E [1] and is self-dual with respect to the Euclidean inner product, $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{v}$. There is no additive variant of these codes, and this family will not be considered in this paper. Again, some constructions have been described [10], but no complete classifications of Euclidean self-dual codes over $\mathrm{GF}(9)$ have been given.

In Section II we briefly review the connection between trace-Hermitian self-dual additive codes and weighted graphs. An algorithm for checking equivalence of self-dual additive codes over $\mathrm{GF}(9)$, which is a generalization of a known algorithm for linear codes [17], is described in Sections III and IV. Combining this algorithm with the weighted graph representation, and some other optimizations, enables us to classify all self-dual additive codes over $\mathrm{GF}(9)$ of length up to 10 in Section V. In particular, all near-extremal codes of length 9 and 10 are classified for the first time. We also find the smallest codes with trivial automorphism group. Using an extension technique described in Section VI, we are then able to classify all optimal codes of length 11 and 12. We finish with some concluding remarks in Section VII.

II. Codes and Weighted Graphs

An *m-weighted graph* is a triple $G = (V, E, W)$, where V is a set of *vertices*, $E \subseteq V \times V$ is a set of *edges*, and W is a set of weights from $\mathrm{GF}(m)$, such that each edge has an associated non-zero weight. In an unweighted graph, which is simply described by a pair $G = (V, E)$, we can consider all edges to have weight one. A graph with n vertices can be represented by an $n \times n$ *adjacency matrix* Γ, where the element $\Gamma_{i,j} = W(\{i, j\})$ if $\{i, j\} \in E$, and $\Gamma_{i,j} = 0$ otherwise. A *loop-free undirected* graph has a symmetric adjacency matrix where all diagonal elements are 0. In a *directed* graph, edges are ordered pairs, and the adjacency matrix is not necessarily symmetric. In a *colored* graph, the set of vertices is partitioned into disjoint subsets, where each subset is assigned a different color. Two graphs $G = (V, E)$ and $G' = (V, E')$ are *isomorphic* if and only if there exists a permutation π of V such that $\{u, v\} \in E \iff \{\pi(u), \pi(v)\} \in E'$. For weighted graphs, we also require that edge weights are preserved, i.e., $W(\{u, v\}) = W(\{\pi(u), \pi(v)\})$. For a colored graph, we further require the permutation to preserve the graph coloring, i.e., that all vertices are mapped to vertices of the same color. The *automorphism group* of a graph is the set of vertex permutations that map the graph to itself. A *path* is a sequence of vertices, $(v_1, v_2, \ldots, v_i)$, such that $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{i-1}, v_i\} \in E$. A graph is *connected* if there is a path from any vertex to any other vertex in the graph.

If an additive code over $\mathrm{GF}(9)$ has a generator matrix of the form $C = \Gamma + \omega I$, where I is the identity matrix, ω is a primitive element of $\mathrm{GF}(9)$, and Γ is the adjacency matrix of a loop-free undirected 3-weighted graph, we say that the generator matrix is in *standard form*. A generator matrix in standard form must generate a code that is self-dual with respect to the Hermitian trace inner product, since it has full rank over $\mathrm{GF}(3)$ and $C\overline{C}^{\mathrm{T}} = \Gamma^2 + \Gamma - I$ only contains entries from $\mathrm{GF}(3)$, and hence the traces of all elements of $\omega^2 C\overline{C}^{\mathrm{T}}$ will be zero.

In the context of quantum codes, it was shown by Schlingemann [18] and by Grassl, Klappenecker, and Rötteler [19] that every self-dual additive code is equivalent to a code with a generator matrix in standard form. Essentially, the same results was also shown by Bouchet [20] in the context of *isotropic systems*. The algorithm given in Fig. 1 can be used to perform a mapping from a self-dual additive code to an equivalent code in standard form. Note that we can write the generator matrix $C = A + \omega B$ as an $n \times 2n$ matrix $(A \mid B)$ with elements from $\mathrm{GF}(3)$. Steps 1 and 2 of the algorithm are used to obtain the submatrices A and B. If B now has full rank, we can simply perform the basis change $B^{-1}(A \mid B) = (\Gamma \mid I)$ to obtain the standard form. Elements on the diagonal of Γ can then always be set to zero by operations $a + b\omega \mapsto a + yb + b\omega$, for $y \in \mathrm{GF}(3)$, corresponding to symplectic matrices $\left(\begin{smallmatrix} 1 & y \\ 0 & 1 \end{smallmatrix}\right)$. Hence step 12 of the algorithm preserves code equivalence. In the case where B has rank $k < n$, we can assume, after a basis change, that the first k rows and columns of B form a $k \times k$ invertible matrix. This is done in step 5, and the result is a permutation of the coordinates of the code. By the operation $c \mapsto \omega\overline{c}$, for $c = a + b\omega$, corresponding to the symplectic matrix $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, we can replace column $\boldsymbol{a_i}$ by $-\boldsymbol{b_i}$ and $\boldsymbol{b_i}$ by $\boldsymbol{a_i}$. In this way, we "swap" the $n-k$ last columns of A and B in steps 7 and 8. It has been shown that it then follows from the self-duality of the code that the new matrix B must have full rank [4], [21], and that the matrix Γ obtained in step 11 will always be symmetric.

As an example, consider the $(4, 3^4, 3)$ code generated by C which by the described algorithm is transformed into the standard form generator matrix C', corresponding to the weighted graph depicted in Fig. 2:

$$C = \begin{pmatrix} 1 & 0 & 1 & \omega^2 \\ \omega & 0 & \omega & \omega^3 \\ 0 & 1 & \omega^2 & 1 \\ 0 & \omega & \omega^3 & \omega \end{pmatrix} \quad C' = \begin{pmatrix} \omega & -1 & 0 & 1 \\ -1 & \omega & 1 & 0 \\ 0 & 1 & \omega & 1 \\ 1 & 0 & 1 & \omega \end{pmatrix}$$

It is known that two self-dual additive codes over $\mathrm{GF}(4)$ are equivalent if and only if their corresponding graphs are related by a sequence of graph operations called *local complementations* (LC) [7], [20], [21] and a permutation of the vertices. We have previously used this fact to devise an algorithm to classify all self-dual additive codes over $\mathrm{GF}(4)$ of

Require: C generates a self-dual additive code over $\mathrm{GF}(9)$.
Ensure: C' generates an equivalent code in standard form.

1: $A \leftarrow \mathrm{Tr}(\omega C)$
2: $B \leftarrow \mathrm{Tr}(\omega^2 C)$
3: $k \leftarrow \mathrm{rank}(B)$
4: **if** $k < n$ **then**
5: Permute rows and columns of B such that the first k rows and columns form an invertible matrix. Apply the same permutation to the rows and columns of A.
6: **for** $i = k+1$ to n **do**
7: Swap columns $\boldsymbol{a_i}$ and $\boldsymbol{b_i}$
8: $\boldsymbol{a_i} \leftarrow -\boldsymbol{a_i}$
9: **end for**
10: **end if**
11: $\Gamma \leftarrow B^{-1}A$
12: Set all diagonal elements of Γ to zero.
13: $C' \leftarrow \Gamma + \omega I$
14: **return** C'

Fig. 1. Algorithm for Mapping a Code to Standard Form

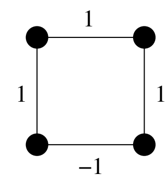

Fig. 2. Graph Representation of the $(4, 3^4, 3)$ Code

length up to 12 [12]. The more general result that equivalence classes of self-dual additive codes over $\mathrm{GF}(q = m^2)$ can be represented as orbits of m-weighted graphs with respect to a generalization of LC was later shown by Bahramgiri and Beigi [22]. We used this to classify all self-dual additive codes over $\mathrm{GF}(9)$, $\mathrm{GF}(16)$, and $\mathrm{GF}(25)$ up to lengths 8, 6, and 6, respectively [4]. The main obstacle with this approach is that the sizes of the LC orbits of weighted graphs quickly get unmanageable as the number of vertices increase. We have therefore devised a new method for checking code equivalence, which is described in the next section. This algorithm uses a graph representation of self-dual additive codes over $\mathrm{GF}(9)$ that is not related to the representation described in this section, and does not require the input to be in standard form. However, the weighted graph representation will still be very useful for reducing our search space.

III. Equivalence Graphs

To check whether two self-dual additive codes over $\mathrm{GF}(9)$ are equivalent, we modify a well-known algorithm used for checking equivalence of linear codes, described by Östergård [17]. The idea is to map a code to an unweighted, directed, colored *equivalence graph* such that the automorphism groups of the code and the equivalence graph coincide. An important component of the algorithm is to find a suitable *coordinate graph*. For self-dual additive codes over $\mathrm{GF}(9)$, we need to construct a graph G on eight vertices, labeled with the

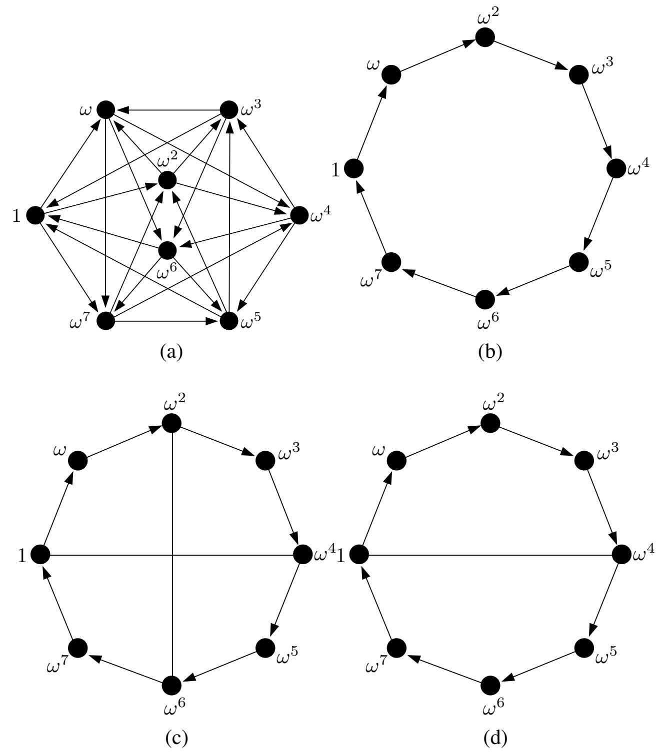

Fig. 3. Coordinate Graphs for Codes over GF(9): (a) Trace-Hermitian Self-Dual Additive (b) Linear (c) Hermitian Self-Dual Linear (d) Euclidean Self-Dual Linear

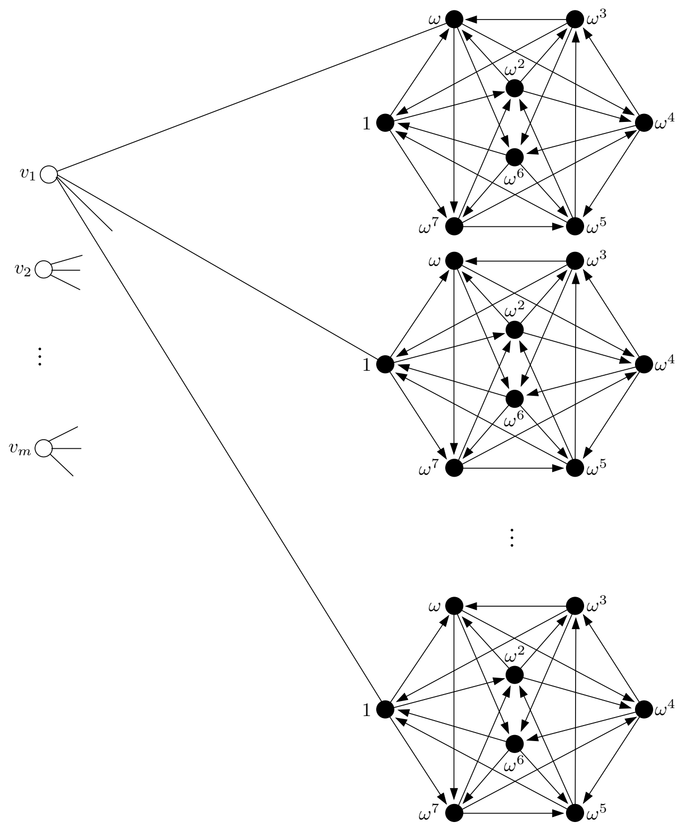

Fig. 4. Example of Equivalence Graph

non-zero elements of $\mathrm{GF}(9)$, whose automorphism group is $\mathrm{Sp}_2(3)$. This graph, shown in Fig. 3 (a), was found by adding directed edges $(\sigma 1, \sigma\omega)$ for all $\sigma \in \mathrm{Sp}_2(3)$. This ensures that $\mathrm{Sp}_2(3) \subseteq \mathrm{Aut}(G)$. We then verified that $|\mathrm{Aut}(G)| = 24$ which implies that $\mathrm{Aut}(G) = \mathrm{Sp}_2(3)$.

Fig. 3 also shows examples of coordinate graphs for some other families of codes over $\mathrm{GF}(9)$. In the original algorithm for checking equivalence of linear codes [17], the coordinate graph shown in Fig. 3 (b) would be used. This graph has an automorphism group of size eight, corresponding to the fact that multiplication of a coordinate by any non-zero element of $\mathrm{GF}(9)$ preserves linearity. The more restrictive coordinate graph for Hermitian self-dual linear codes over $\mathrm{GF}(9)$ is shown in Fig. 3 (c). This graph has an automorphism group of size four, since only multiplication by $x \in \mathrm{GF}(9)$ where $x^4 = 1$ is permitted in this case. Finally, Fig. 3 (d) shows a graph with automorphism group of size two. This is the coordinate graph for Euclidean self-dual linear codes over $\mathrm{GF}(9)$ where multiplication by ± 1 are the only permitted operations. Coordinate graphs of this type were used by Harada and Östergård to classify Euclidean self-dual codes over $\mathrm{GF}(5)$ up to length 16 [23] and over $\mathrm{GF}(7)$ up to length 12 [24].

To construct the equivalence graph of a code, we first add n copies of the coordinate graph, each copy representing one coordinate of the code. We then need a deterministic way to find a set of codewords that generates the code. Taking all codewords would suffice, but the following approach yields a smaller set and hence a more efficient algorithm. First, we check if the set of all codewords of minimum weight d generates the code. If it does not, we add all codewords of weight $d+1$, then all codewords of weight $d+2$, etc, stopping once we have a set that spans the code. For each codeword c_i in the resulting set, we add a *codeword vertex* v_i to the equivalence graph. Let the codeword vertices have one color, and the other vertices have a different color. Edges are added between v_i and the coordinate graphs according to the non-zero coordinates of the codeword c_i, e.g., if c_i has ω in coordinate j, then there is an edge between v_i and the vertex labeled ω in the jth coordinate graph. As an example, Fig. 4 shows the case where $c_1 = (\omega 1 \cdots 1)$. The resulting equivalence graph is finally *canonized*, i.e., relabeled, but with coloring preserved, using the *nauty* software [25]. If two graphs are isomorphic, their canonical representations are guaranteed to be the same.

Applying a canonical permutation to the vertices of an equivalence graph corresponds to permuting the coordinates of the corresponding code, applying elements from $\mathrm{Sp}_2(3)$ to each coordinate, and sorting the codewords c_i in some canonical order. If two codes are equivalent, their canonical equivalence graphs will therefore be identical. Furthermore, the automorphism group of a code is equivalent to the automorphism group of its equivalence graph. This follows from the fact that any automorphism of the equivalence graph must be one out of $24^n n!$ possibilities, i.e., the $n!$ permutations of the n coordinate subgraphs, and the 24 automorphisms from $\mathrm{Sp}_2(3)$ of each coordinate subgraph. No other automorphisms are possible. In particular, permuting the codeword vertices will never be an automorphism, since all codewords must be distinct. Since it is known [2] that coordinate permutations and $\mathrm{Sp}_2(3)$ applied to the coordinates of a code preserve its weight enumerator, additivity, and self-duality, this must also be true for any automorphism of the equivalence graph.

IV. Classification Algorithm

We have seen that every weighted graph corresponds to a self-dual additive code, and that every self-dual additive code, up to equivalence, has a standard form representation as a weighted graph. It follows that we only need to consider 3-weighted graphs in order to classify all self-dual additive codes over $\mathrm{GF}(9)$. Permuting the vertices of a graph corresponds to permuting coordinates of the associated code, which means that we only need to consider these graphs up to isomorphism. Moreover, we can restrict our study to connected graphs, since a disconnected graph represents a *decomposable* code. A code is decomposable if it can be written as the *direct sum* of two smaller codes. For example, let $\mathcal{C}$ be an $(n, 3^n, d)$ code and $\mathcal{C}'$ an $(n', 3^{n'}, d')$ code. The direct sum, $\mathcal{C} \oplus \mathcal{C}' = \{u||v \mid u \in \mathcal{C}, v \in \mathcal{C}'\}$, where $||$ means concatenation, is an $(n+n', 3^{n+n'}, \min\{d, d'\})$ code. All decomposable codes of length n can be generated easily once all indecomposable codes of length less than n are known.

To classify codes of length n, we could take all non-isomorphic connected 3-weighted graphs on n vertices, map the corresponding codes to equivalence graphs, and canonize these. All duplicates would then be removed to obtain one representative code from each equivalence class. However, a much smaller set of graphs is obtained by taking all possible *lengthenings* [26] of all codes of length $n-1$. A generator matrix in standard form can be lengthened in $3^{n-1}-1$ ways by adding a vertex to the corresponding graph and connecting it to all possible combinations of at least one of the old vertices, using all possible combinations of edge weights. This corresponds to adding a new non-zero row $\boldsymbol{r} \in \mathrm{GF}(3)^n$ and column $\boldsymbol{r}^{\mathrm{T}}$ to the adjacency matrix, with zero in the last coordinate. Only half of the lengthenings need to be considered, as adding the row $-\boldsymbol{r}$ is equivalent to adding $\boldsymbol{r}$. (Since multiplying the last row and column in the corresponding generator matrix by -1 would preserve code equivalence.) We have previously shown [4], using the theory of local complementation of weighted graphs, that the set of $i_{n-1}\frac{3^{n-1}-1}{2}$ codes obtained by lengthening one representative from each of the i_{n-1} equivalence classes of indecomposable codes of length $n-1$ must contain at least one representative from each equivalence class of the indecomposable codes of length n.

Removing possible isomorphisms from the set of lengthened graphs, using *nauty* [25], speeds up our classification significantly. A set of non-isomorphic graphs that have already been processed, as large as memory resources permit, can even be stored between iterations, and new graphs can be checked for isomorphism against this set. For each graph that is not excluded by such an isomorphism check, the corresponding code must be mapped to an equivalence graph, as described in Section III. The equivalence graph is canonized and compared against all previously observed codes, which are stored in memory. Since the equivalence graphs will be large, typically containing thousands of vertices for $n = 10$, we map the equivalence graph to a canonical generator matrix by taking the first n linearly independent codewords corresponding to codeword vertices in their canonical ordering. This generator matrix can further be mapped to a canonical standard form, as described in Section II, which means that only $\binom{n}{2}$ ternary symbols need to be stored for each code. An outline of the steps of our classification algorithm is listed in Fig. 5.

Require: $\boldsymbol{C}_{n-1}$ contains one graph representation of each inequivalent indecomposable code of length $n-1$.
Ensure: $\boldsymbol{C}_n$ contains one graph representation of each inequivalent indecomposable code of length n.

$\boldsymbol{C}_n \leftarrow \emptyset$
for all $C \in \boldsymbol{C}_{n-1}$ **do**
  $\boldsymbol{E} \leftarrow \frac{3^{n-1}-1}{2}$ lengthenings of C
  Remove isomorphisms from $\boldsymbol{E}$
  for all $E \in \boldsymbol{E}$ **do**
    $d \leftarrow$ minimum distance of E
    $\boldsymbol{S} \leftarrow$ all codewords of weight d from E
    while $\boldsymbol{S}$ does not generate E **do**
      $d \leftarrow d + 1$
      $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup$ all codewords of weight d from E
    end while
    $Q \leftarrow$ equivalence graph given by $\boldsymbol{S}$
    $Q' \leftarrow$ canonize Q
    $G \leftarrow$ graph representation of code given by Q'
    if $G \notin \boldsymbol{C}_n$ **then**
      $\boldsymbol{C}_n \leftarrow \boldsymbol{C}_n \cup G$
    end if
  end for
end for
return $\boldsymbol{C}_n$

Fig. 5. Classification Algorithm

Note that the special form of a generator matrix in standard form makes it easy to find all codewords of low weight, which is necessary to construct the equivalence graph. If $\mathcal{C}$ is generated by $C = \Gamma + \omega I$, then any codeword formed by taking $\mathrm{GF}(3)$-linear combinations of i rows of C must have weight at least i. This means that we can find all codewords of weight i by only considering combinations of at most i rows of C.

V. Codes of Length 9 and 10

Using the algorithm described in Section IV, we have classified all self-dual additive codes over $\mathrm{GF}(9)$ of length up to 10. Table I gives the values of i_n, the number of inequivalent indecomposable codes of length n, and the values of t_n, the total number of inequivalent codes of length n. Note that the numbers t_n are easily derived from the numbers i_n by using the *Euler transform* [27]:

$$\begin{aligned} c_n &= \sum_{d|n} d i_d \\ t_1 &= c_1 \end{aligned}$$

TABLE I
NUMBER OF INDECOMPOSABLE (i_n) AND TOTAL NUMBER (t_n) OF SELF-DUAL ADDITIVE CODES OVER GF(9) OF LENGTH n

n	1	2	3	4	5	6	7	8	9	10
i_n	1	1	1	3	5	21	73	659	17 589	2 803 404
t_n	1	2	3	7	13	39	121	817	18 525	2 822 779

$$t_n = \frac{1}{n}\left(c_n + \sum_{k=1}^{n-1} c_k t_{n-k}\right).$$

Tables II and III list the numbers of indecomposable codes and the total number of codes, respectively, by length and minimum distance. In Table IV, we count the number of distinct weight enumerators. There are obviously too many codes of length 9 and 10 to list all of them here, so an on-line database containing one representative from each equivalence class has been made available at http://www.ii.uib.no/~larsed/nonbinary/.

Generator matrices for all the extremal codes of length 9 and 10 were given in [4]. Our classification confirms that there are four extremal $(9, 3^9, 5)$ codes, all with weight enumerator $W(y) = 1 + 252y^5 + 1176y^6 + 3672y^7 + 7794y^8 + 6788y^9$, and with automorphism groups of size 72, 108, 108, and 432, and that there is a unique extremal $(10, 3^{10}, 6)$ code with weight enumerator $W(y) = 1 + 1680y^6 + 2880y^7 + 14040y^8 + 22160y^9 + 18288y^{10}$ and automorphism group of size 2880. The classification of near-extremal codes of length 9 and 10 is new. We find that there are 4370 near-extremal $(9, 3^9, 4)$ codes with 25 distinct weight enumerators and 13 different values for $|\mathrm{Aut}(\mathcal{C})|$. The weight enumerators that exist are given by $W_{9,\alpha}(y) = 1 + (4+2\alpha)y^4 + (244-4\alpha)y^5 + (1168-4\alpha)y^6 + (3704+16\alpha)y^7 + (7766-14\alpha)y^8 + (6796+4\alpha)y^9$ for all integer values $0 \le \alpha \le 24$. Table V gives the number of $(9, 3^9, 4)$ codes for each possible weight enumerator and automorphism group size. To highlight a few codes with extreme properties, we list generator matrices for the code with automorphism group of maximal size (288) and one of the codes with weight enumerator $W_{9,0}(y)$, i.e., with the minimal number of minimum weight codewords. For the latter case, we choose the unique code with maximum number of automorphisms (16). In the following, "-" denotes -1 in generator matrices:

$$C^{n=9}_{|\mathrm{Aut}|=288} = \begin{pmatrix}
\omega & \text{-} & 1 & 1 & \text{-} & \text{-} & 1 & \text{-} & 1 \\
\text{-} & \omega & 1 & 1 & 1 & 0 & 1 & \text{-} & 0 \\
1 & 1 & \omega & 1 & 1 & 0 & 0 & 1 & \text{-} \\
1 & 1 & 1 & \omega & 1 & 1 & \text{-} & 0 & 0 \\
\text{-} & 1 & 1 & 1 & \omega & \text{-} & 0 & 0 & 1 \\
\text{-} & 0 & 0 & 1 & \text{-} & \omega & 0 & 1 & 0 \\
1 & 1 & 0 & \text{-} & 0 & 0 & \omega & 0 & 1 \\
\text{-} & \text{-} & 1 & 0 & 0 & 1 & 0 & \omega & 0 \\
1 & 0 & \text{-} & 0 & 1 & 0 & 1 & 0 & \omega
\end{pmatrix}$$

$$C^{n=9}_{\alpha=0,\,|\mathrm{Aut}|=16} = \begin{pmatrix}
\omega & 1 & 1 & 1 & 1 & \text{-} & \text{-} & \text{-} & \text{-} \\
1 & \omega & 1 & \text{-} & \text{-} & 1 & \text{-} & 1 & 0 \\
1 & 1 & \omega & \text{-} & \text{-} & \text{-} & 1 & 0 & 1 \\
1 & \text{-} & \text{-} & \omega & 1 & 0 & 1 & \text{-} & 1 \\
1 & \text{-} & \text{-} & 1 & \omega & 1 & 0 & 1 & \text{-} \\
\text{-} & 1 & \text{-} & 0 & 1 & \omega & \text{-} & \text{-} & 1 \\
\text{-} & \text{-} & 1 & 1 & 0 & \text{-} & \omega & 1 & \text{-} \\
\text{-} & 1 & 0 & \text{-} & 1 & \text{-} & 1 & \omega & \text{-} \\
\text{-} & 0 & 1 & 1 & \text{-} & 1 & \text{-} & \text{-} & \omega
\end{pmatrix}$$

We find that there are 4577 near-extremal $(10, 3^{10}, 5)$ codes with 10 distinct weight enumerators and 20 different values for $|\mathrm{Aut}(\mathcal{C})|$. The weight enumerators that exist are given by $W_{10,\alpha}(y) = 1 + (44 + 4\alpha)y^5 + (1460 - 20\alpha)y^6 + (3320 + 40\alpha)y^7 + (13600 - 40\alpha)y^8 + (22380 + 20\alpha)y^9 + (18244 - 4\alpha)y^{10}$ for integer values $\alpha \in \{0, 9, 12, 13, 16, 18, 21, 22, 24, 25\}$. Table VI gives the number of $(10, 3^{10}, 5)$ codes for each possible weight enumerator and automorphism group size. We give generator matrices for the unique codes with automorphism groups of size 2880 and 288, as well as the unique code with weight enumerator $W_{10,0}(y)$:

$$C^{n=10}_{|\mathrm{Aut}|=2880} = \begin{pmatrix}
\omega & 1 & 1 & 1 & 1 & 0 & \text{-} & \text{-} & 1 & 1 \\
1 & \omega & 1 & 1 & 1 & 1 & \text{-} & 1 & \text{-} & 0 \\
1 & 1 & \omega & 1 & 1 & 1 & 1 & \text{-} & 0 & \text{-} \\
1 & 1 & 1 & \omega & 1 & \text{-} & 0 & 1 & 1 & \text{-} \\
1 & 1 & 1 & 1 & \omega & \text{-} & 1 & 0 & \text{-} & 1 \\
0 & 1 & 1 & \text{-} & \text{-} & \omega & 1 & 1 & 1 & 1 \\
\text{-} & \text{-} & 1 & 0 & 1 & 1 & \omega & 1 & 1 & 1 \\
\text{-} & 1 & \text{-} & 1 & 0 & 1 & 1 & \omega & 1 & 1 \\
1 & \text{-} & 0 & 1 & \text{-} & 1 & 1 & 1 & \omega & 1 \\
1 & 0 & \text{-} & \text{-} & 1 & 1 & 1 & 1 & 1 & \omega
\end{pmatrix}$$

$$C^{n=10}_{|\mathrm{Aut}|=288} = \begin{pmatrix}
\omega & 0 & 0 & \text{-} & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & \omega & \text{-} & \text{-} & 1 & 0 & 1 & 0 & \text{-} & 1 \\
0 & \text{-} & \omega & \text{-} & 1 & 0 & 0 & 1 & 1 & \text{-} \\
\text{-} & \text{-} & \text{-} & \omega & 1 & 0 & \text{-} & \text{-} & 1 & 1 \\
1 & 1 & 1 & 1 & \omega & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \omega & 1 & \text{-} & \text{-} & 1 \\
0 & 1 & 0 & \text{-} & 0 & 1 & \omega & \text{-} & 1 & 0 \\
0 & 0 & 1 & \text{-} & 0 & \text{-} & \text{-} & \omega & 0 & 1 \\
1 & \text{-} & 1 & 1 & 0 & \text{-} & 1 & 0 & \omega & \text{-} \\
1 & 1 & \text{-} & 1 & 0 & 1 & 0 & 1 & \text{-} & \omega
\end{pmatrix}$$

$$C^{n=10}_{\alpha=0} = \begin{pmatrix}
\omega & 1 & 0 & \text{-} & \text{-} & 1 & 0 & \text{-} & 0 & 0 \\
1 & \omega & 1 & 1 & 0 & 1 & 0 & \text{-} & 1 & \text{-} \\
0 & 1 & \omega & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\text{-} & 1 & 0 & \omega & 0 & \text{-} & 1 & \text{-} & \text{-} & \text{-} \\
\text{-} & 0 & 1 & 0 & \omega & 1 & \text{-} & \text{-} & \text{-} & \text{-} \\
1 & 1 & 1 & \text{-} & 1 & \omega & 1 & \text{-} & 1 & 0 \\
0 & 0 & 0 & 1 & \text{-} & 1 & \omega & 1 & \text{-} & 0 \\
\text{-} & \text{-} & 1 & \text{-} & \text{-} & \text{-} & 1 & \omega & 1 & 1 \\
0 & 1 & 1 & \text{-} & \text{-} & 1 & \text{-} & 1 & \omega & 1 \\
0 & \text{-} & 0 & \text{-} & \text{-} & 0 & 0 & 1 & 1 & \omega
\end{pmatrix}$$

That our classification of all codes up to length 10 is correct has been verified by the mass formula (1). This required us to also calculate the sizes of the automorphism groups of all decomposable codes, which was simplified by the observation that for a code $\mathcal{C} = k_1\mathcal{C}_1 \oplus \cdots \oplus k_m\mathcal{C}_m$, where $k_j\mathcal{C}_j = \bigoplus_{i=1}^{k_j} \mathcal{C}_j$, $|\mathrm{Aut}(\mathcal{C})| = \prod_{i=1}^{m} k_i!\,|\mathrm{Aut}(\mathcal{C}_i)|^{k_i}$.

Table VII gives the numbers of codes with trivial automorphism group by length and minimum distance. We find that the smallest codes with trivial automorphism group are 35 codes of length 8. (Note that automorphism group sizes were not calculated in the previous classification of codes of length 8 [4].) We give the generator matrix for one $(8, 3^8, 4)$ code with trivial automorphism group. Generator matrices for the other codes can be obtained from http://www.ii.uib.no/~larsed/nonbinary/.

TABLE II
NUMBER OF INDECOMPOSABLE SELF-DUAL ADDITIVE CODES OVER GF(9) OF LENGTH n AND MINIMUM DISTANCE d

$d\backslash n$	2	3	4	5	6	7	8	9	10	11	12
2	1	1	2	4	15	51	388	6240	418 088	?	?
3			1	1	5	20	194	6975	893 422	?	?
4					1	2	77	4370	1 487 316	?	?
5								4	4577	56 005 876	?
6									1		6493
All	1	1	3	5	21	73	659	17 589	2 803 404	?	?

TABLE III
TOTAL NUMBER OF SELF-DUAL ADDITIVE CODES OVER GF(9) OF LENGTH n AND MINIMUM DISTANCE d

$d\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	7	13	39	121	817	18 525	2 822 779	?
2		1	1	3	5	20	60	424	6358	418 931	?	?
3				1	1	5	20	195	6976	893 429	?	?
4						1	2	77	4370	1 487 316	?	?
5									4	4577	56 005 876	?
6										1		6493
All	1	2	3	7	13	39	121	817	18 525	2 822 779	$> 2^{30}$	$> 2^{41}$

TABLE IV
NUMBER OF DISTINCT WEIGHT ENUMERATORS OF INDECOMPOSABLE CODES OF LENGTH n AND MINIMUM DISTANCE d

$d\backslash n$	2	3	4	5	6	7	8	9	10	11	12
2	1	1	2	4	14	42	202	1021	8396	?	?
3			1	1	3	9	33	170	1133	?	?
4					1	1	9	25	345	?	?
5								1	10	48	?
6									1		27
All	1	1	3	5	18	52	244	1217	9885	?	?

$$C^{n=8}_{|\mathrm{Aut}|=2} = \begin{pmatrix} \omega & 0 & 0 & 1 & 1 & \text{-} & \text{-} & \text{-} \\ 0 & \omega & \text{-} & 0 & 1 & 1 & 1 & \text{-} \\ 0 & \text{-} & \omega & \text{-} & 1 & 0 & \text{-} & 0 \\ 1 & 0 & \text{-} & \omega & 0 & \text{-} & 0 & 1 \\ 1 & 1 & 1 & 0 & \omega & 1 & 0 & 0 \\ \text{-} & 1 & 0 & \text{-} & 1 & \omega & 0 & 0 \\ \text{-} & 1 & \text{-} & 0 & 0 & 0 & \omega & \text{-} \\ \text{-} & \text{-} & 0 & 1 & 0 & 0 & \text{-} & \omega \end{pmatrix}$$

We observe that codes with minimum distance $d \leq 2$ always have nontrivial automorphisms, and this can be proved as follows. For $d = 1$, we can assume that the first row of a standard form generator matrix is $(\omega 0 \cdots 0)$. Then $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ applied to the first coordinate of the code is an automorphism of order 3. Multiplying the first coordinate by -1 has the same effect as multiplying the first row by -1 and is therefore an automorphism of order 2. Including the trivial automorphism, we have that $|\mathrm{Aut}| \geq 12$. There are codes of length 9 with $d = 1$ and $|\mathrm{Aut}| = 12$ which shows that the bound is tight. For $d = 2$, we can assume that the first row of a standard form generator matrix is $(\omega 1 0 \cdots 0)$. Then $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ applied to the first coordinate and $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ applied to the second coordinate of the code has the same effect as adding the first row of the generator matrix to the second row, and is hence an automorphism of order 3. Swapping the first two coordinates is an automorphism of order 2, since it has the same effect as the following procedure: Add the first row to itself, then add the first row to each row $i > 2$ where the value in position i of the second column is 1, and add twice the first row to each row $i > 2$ where the value in position i of the second column is 2. Finally apply $\left(\begin{smallmatrix} 0 & 2 \\ 1 & 0 \end{smallmatrix}\right)$ to the first column, and $\left(\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right)$ to the second column. Again, including the trivial automorphism we get the bound $|\mathrm{Aut}| \geq 12$, and the existence of codes of length 8 with $d = 2$ and $|\mathrm{Aut}| = 12$ proves that the bound is tight.

VI. OPTIMAL CODES OF LENGTH 11 AND 12

When we lengthen an $(n, 3^n, d)$ code, as described in Section IV, we always obtain an $(n+1, 3^{n+1}, d')$ code where $d' \leq d + 1$ [26]. It follows that given a classification of all codes of length n and minimum distance at least d, we can classify all codes of length $n+1$ and minimum distance at least $d+1$. There are no $(11, 3^{11}, 6)$ codes, but by lengthening the 1 491 894 $(10, 3^{10}, d)$ codes for $d \geq 4$, we are able to obtain all optimal $(11, 3^{11}, 5)$ codes. To quickly exclude codes with $d < 5$, we checked the minimum distance of each lengthened code before checking for code equivalence in this search.

TABLE V

NUMBER OF $(9, 3^9, 4)$ CODES WITH WEIGHT ENUMERATOR $W_{9,\alpha}(y)$ AND $|\mathrm{Aut}(\mathcal{C})| = \beta$

$\alpha\backslash\beta$	2	4	6	8	12	16	24	32	36	48	72	144	288	All
0		3		1		1								5
1	2	1	2											5
2	15	21		4										40
3	15	13	1	3	2		1							35
4	125	52			12	2	2			2				195
5	85	8												93
6	338	93		11	2	1								445
7	165	53	2	9	2	2	2				1			236
8	561	150		11				1						723
9	173	20	6	7			1							207
10	522	154	4	7	7		2			2				698
11	157	53		15										225
12	356	143	2	4	3	2								510
13	119	25	2	6										152
14	229	114		11		2								356
15	42	28	1	16	1	2								90
16	96	62		8	2	3	2		4			2	1	180
17	15	9		6										30
18	23	33		2	1	6		2						67
19	9	4		6		2	2							23
20	8	23		6				2						39
21		2	2				1							5
22	1	3				2		1						7
23				1										1
24								3						3
All	3056	1067	22	134	32	25	13	9	4	4	1	2	1	4370

TABLE VI

NUMBER OF $(10, 3^{10}, 5)$ CODES WITH WEIGHT ENUMERATOR $W_{10,\alpha}(y)$ AND $|\mathrm{Aut}(\mathcal{C})| = \beta$

$\alpha\backslash\beta$	2	4	6	8	10	12	16	20	24	32	36	40	48	64	72	144	192	240	288	2880	All
0																		1			1
9							1					1					1	1			4
12							2														2
13		3					3			1											7
16	10	5		2							1										18
18	30	24	4	4		8	4														74
21	190	77	2	20		2	2		4					3							300
22	467	72		4			1														544
24	2321	172	4	4	1	5	1					1	1								2510
25	777	247	12	39		14	10	3	2	1	2	2		4	1	1			1	1	1117
All	3795	600	22	73	1	29	24	3	6	2	3	4	1	7	1	1	1	2	1	1	4577

TABLE VII
NUMBER OF CODES OF LENGTH n AND MINIMUM DISTANCE d WITH TRIVIAL AUTOMORPHISM GROUP

$d\backslash n$	≤ 7	8	9	10	11	12
≤ 2	0	0	0	0	0	0
3	0	32	4518	832 878	?	?
4	0	3	3056	1 419 861	?	?
5			0	3795	55 865 753	?
6				0		3445
All	0	35	7574	2 256 534	?	?

We find that there are 56 005 876 optimal $(11, 3^{11}, 5)$ codes with 48 distinct weight enumerators and 24 different values for $|\mathrm{Aut}(\mathcal{C})|$. The weight enumerators that exist are given by $W_{11,\alpha}(y) = 1 + (12 + 2\alpha)y^5 + (888 - 6\alpha)y^6 + 3960y^7 + (14970 + 20\alpha)y^8 + (42500 - 30\alpha)y^9 + (66240 + 18\alpha)y^{10} + (48756 - 4\alpha)y^{11}$ for all integer values $6 \leq \alpha \leq 50$ as well as $\alpha \in \{0, 54, 60\}$. Observe that the number of codewords of weight 7 is constant for all codes. Table VIII gives the number of $(11, 3^{11}, 5)$ codes for each possible weight enumerator and automorphism group size. We give generator matrices for the unique codes with automorphism group of size 47 520 and 1440, as well as the unique code with weight enumerator $W_{11,0}(y)$:

$$C^{n=11}_{|\mathrm{Aut}|=47\,520} = \begin{pmatrix} \omega & 1 & - & 1 & 1 & - & 1 & 1 & 1 & 0 & 0 \\ 1 & \omega & 1 & 0 & 1 & - & - & - & 1 & 0 & 1 \\ - & 1 & \omega & 0 & - & 0 & - & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & \omega & 0 & 1 & 1 & - & - & 0 & 1 \\ 1 & 1 & - & 0 & \omega & 0 & 0 & 0 & 1 & - & 0 \\ - & - & 0 & 1 & 0 & \omega & 0 & 0 & 1 & - & 0 \\ 1 & - & - & 1 & 0 & 0 & \omega & 0 & - & 0 & 0 \\ 1 & - & 1 & - & 0 & 0 & 0 & \omega & 0 & - & 0 \\ 1 & 1 & 1 & - & 1 & 1 & - & 0 & \omega & - & - \\ 0 & 0 & 1 & 0 & - & - & 0 & - & - & \omega & - \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & - & - & \omega \end{pmatrix}$$

$$C^{n=11}_{|\mathrm{Aut}|=1440} = \begin{pmatrix} \omega & 1 & - & 1 & - & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & \omega & - & 1 & 1 & - & - & 0 & 1 & 1 & 0 \\ - & - & \omega & 1 & 0 & 1 & - & - & - & 0 & - \\ 1 & 1 & 1 & \omega & 0 & - & - & - & 0 & - & 1 \\ - & 1 & 0 & 0 & \omega & 1 & - & 1 & 1 & - & 1 \\ 0 & - & 1 & - & 1 & \omega & - & 1 & - & 1 & - \\ 0 & - & - & - & - & - & \omega & 0 & 0 & 0 & 0 \\ 0 & 0 & - & - & 1 & 1 & 0 & \omega & 0 & 0 & 1 \\ 0 & 1 & - & 0 & 1 & - & 0 & 0 & \omega & 0 & - \\ 0 & 1 & 0 & - & - & 1 & 0 & 0 & 0 & \omega & - \\ 1 & 0 & - & 1 & 1 & - & 0 & 1 & - & - & \omega \end{pmatrix}$$

$$C^{n=11}_{\alpha=0} = \begin{pmatrix} \omega & 1 & 0 & - & 1 & 0 & - & 1 & 0 & 0 & 0 \\ 1 & \omega & 1 & 0 & - & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & \omega & 1 & - & 1 & 0 & 0 & 0 & 1 & 0 \\ - & 0 & 1 & \omega & - & - & 0 & - & 0 & - & - \\ 1 & - & - & - & \omega & - & - & 0 & 0 & 0 & 1 \\ 0 & - & 1 & - & - & \omega & - & - & - & 1 & 0 \\ - & 1 & 0 & 0 & - & - & \omega & 0 & - & 0 & 1 \\ 1 & 0 & 0 & - & 0 & - & 0 & \omega & - & - & - \\ 0 & 0 & 0 & 0 & 0 & - & - & - & \omega & 1 & 1 \\ 0 & 0 & 1 & - & 0 & 1 & 0 & - & 1 & \omega & 0 \\ 0 & 0 & 0 & - & 1 & 0 & 1 & - & 1 & 0 & \omega \end{pmatrix}$$

We find that there are 6493 optimal $(12, 3^{12}, 6)$ codes with 27 distinct weight enumerators and 32 different values for $|\mathrm{Aut}(\mathcal{C})|$. The weight enumerators that exist are given by $W_{12,\alpha}(y) = 1 + (480 + 4\alpha)y^6 + (3456 - 24\alpha)y^7 + (15120 + 60\alpha)y^8 + (55520 - 80\alpha)y^9 + (133920 + 60\alpha)y^{10} + (19536 - 24\alpha)y^{11} + (129408 + 4\alpha)y^{12}$ for all integer values $\alpha \in \{0, 1, 3, 4, 7, 9, 12, 13, 16, 19, 21, 25, 27, 28, 31, 36, 37, 39, 43, 48, 49, 52, 57, 63, 64, 81, 144\}$. Table IX gives the number of $(12, 3^{12}, 6)$ codes for each possible weight enumerator and automorphism group size. Generator matrices for all the optimal codes of length 12 can be obtained from http://www.ii.uib.no/~larsed/nonbinary/. We here list generator matrices for the unique code with maximal automorphism group size (2 280 960) and a code with weight enumerator $W_{12,0}(y)$. In the latter case, we choose the single code with maximal number of automorphisms (11 520).

$$C^{n=12}_{|\mathrm{Aut}|=2\,280\,960} = \begin{pmatrix} \omega & - & - & - & - & - & 1 & 0 & 1 & - & - & 0 \\ - & \omega & - & - & - & - & 1 & - & - & 0 & 1 & 0 \\ - & - & \omega & - & - & - & - & - & 1 & 1 & 0 & 0 \\ - & - & - & \omega & - & - & 0 & 1 & - & 1 & - & 0 \\ - & - & - & - & \omega & - & - & 1 & 0 & - & 1 & 0 \\ - & - & - & - & - & \omega & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & - & 0 & - & 0 & \omega & 1 & 1 & 1 & 1 & - \\ 0 & - & - & 1 & 1 & 0 & 1 & \omega & 1 & 1 & 1 & - \\ 1 & - & 1 & - & 0 & 0 & 1 & 1 & \omega & 1 & 1 & - \\ - & 0 & 1 & 1 & - & 0 & 1 & 1 & 1 & \omega & 1 & - \\ - & 1 & 0 & - & 1 & 0 & 1 & 1 & 1 & 1 & \omega & - \\ 0 & 0 & 0 & 0 & 0 & 0 & - & - & - & - & - & \omega \end{pmatrix}$$

$$C^{n=12}_{\alpha=0,\,|\mathrm{Aut}|=11\,520} = \begin{pmatrix} \omega & - & - & - & - & - & 1 & - & - & 1 & 0 & 0 \\ - & \omega & - & - & - & - & 0 & - & 0 & 1 & - & 1 \\ - & - & \omega & - & - & - & 0 & 0 & - & 1 & 1 & - \\ - & - & - & \omega & - & - & 1 & 0 & - & 0 & - & 1 \\ - & - & - & - & \omega & - & 1 & - & 0 & 0 & 1 & - \\ - & - & - & - & - & \omega & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & \omega & - & - & - & 1 & 1 \\ - & - & 0 & 0 & - & 0 & - & \omega & - & - & 1 & 1 \\ - & 0 & - & - & 0 & 0 & - & - & \omega & - & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & - & - & - & \omega & 1 & 1 \\ 0 & - & 1 & - & 1 & 0 & 1 & 1 & 1 & 1 & \omega & 1 \\ 0 & 1 & - & 1 & - & 0 & 1 & 1 & 1 & 1 & 1 & \omega \end{pmatrix}$$

VII. CONCLUSION

According to the mass formula bound (2), the total number of codes of length 11 and 12 are $t_{11} \geq 1\,592\,385\,579$ and $t_{12} \geq 2\,938\,404\,780\,748$, which makes complete classifications infeasible, at least with our computational resources. Running our algorithm on a typical desktop computer, the classification of codes of length n was completed in less than five minutes for $n \leq 8$, about two hours for $n = 9$, and about a week for $n = 10$. Most of this time is spent canonizing the equivalence graphs with *nauty*, and far more time is used on codes with large automorphism groups than on codes with trivial or small automorphism groups. This means that our previous classification algorithm [4], using local complementation, might still be useful in some cases, since we observe that graphs corresponding to codes with large automorphism groups typically have small LC orbits. For instance, we could speed up our classification algorithm by not only removing isomorphisms from the set of lengthened codes, but also generating and storing a limited number of LC orbit members of each graph, and checking new graphs for isomorphism against this set. Finding

TABLE VIII
NUMBER OF $(11, 3^{11}, 5)$ CODES WITH WEIGHT ENUMERATOR $W_{11,\alpha}(y)$ AND $|\mathrm{Aut}(\mathcal{C})| = \beta$

$\alpha\backslash\beta$	2	4	6	8	10	12	16	18	20	24	32	36	40	44	48	72	108	120	144	288	360	432	1440	47 520	All
0								1																	1
6			4					1				3				2									10
7		4		1						2															7
8	10	5																							15
9	36	22	4	4		4						2													72
10	35	16	2	2		2																			57
11	286	62		2																					350
12	217	37	15	7		2	1																		279
13	1515	170	6	8		4				2															1705
14	1140	139		4																					1283
15	7412	414	10	20		14			1			8				5									7884
16	5234	192	4	13			2																		5445
17	30 825	906		28																					31 759
18	19 468	623	17	14		12																			20 134
19	108 109	1606	26	24		8																			109 773
20	62 364	641		7		5																			63 017
21	314 156	2701	16	42		27				2															316 944
22	169 270	1928	4	30		10																			171 242
23	780 271	4123	5	40																					784 439
24	385 400	1508	38	32	1	8	2		1		2	3				1			1	1			1		386 999
25	1 649 942	5666	42	33		2																			1 655 685
26	754 931	4249		44		10	3			4															759 241
27	2 990 527	7882	61	36		44		2				2		6											2 998 560
28	1 266 193	2610	20	19		6					1				2										1 268 851
29	4 671 482	9256	18	36									2												4 680 794
30	1 832 724	6641	41	50		20																			1 839 476
31	6 241 827	10 336	98	39		10																			6 252 310
32	2 266 449	3048		45			6																		2 269 548
33	7 110 043	10 986	89	27		47						7				4	2								7 121 205
34	2 377 017	7970	44	66		4	6																		2 385 107
35	6 821 413	10 684	6	22																					6 832 125
36	2 084 454	3159	46	29		8																			2 087 696
37	5 388 851	9356	99	22		6																			5 398 334
38	1 475 547	6545		30		6																			1 482 128
39	3 403 383	7317	65	27	2	36												1							3 410 831
40	810 399	2084	34	48		2	4						2												812 573
41	1 645 374	5231		13																					1 650 618
42	334 536	3308	35	39		28						3				1	2		1						337 953
43	579 338	2764	32	6		6				2															582 148
44	94 833	664		10			2																		95 509
45	137 174	1487	18			26						2													138 707
46	21 818	713	4			4																			22 539
47	18 178	353																							18 531
48	1901	113	12	2		2	2			1															2033
49	1275	174	3		1	4			2																1459
50	392	80		4																					476
54	4	9		4																					17
60						2									2						1	1		1	7
All	55 865 753	137 782	918	929	4	369	28	4	4	13	3	30	4	6	4	13	4	1	2	1	1	1	1	1	56 005 876

TABLE IX
NUMBER OF $(12, 3^{12}, 6)$ CODES WITH $|\mathrm{Aut}(\mathcal{C})| = \beta$ AND WEIGHT ENUMERATOR $W_{12,\alpha}(y)$

$\beta \backslash \alpha$	0	1	3	4	7	9	12	13	16	19	21	25	27	28	31	36	37	39	43	48	49	52	57	63	64	81	144	All
2	55	117	54	120	186	209	158	338	325	448	418	236	160	268	162	57	86	36	6		6							3445
4	62	135	102	147	184	85	124	214	161	188	222	113	90	134	88	56	48	36	24	7	16							2236
6		5	2	6	2	6			4		8	2	4		2	4					1							46
8	23	25	8	31	40	16	32	28	28	22	18	14	14	26	16	15	6	4			8	2						376
12	1	6	12	10	4	19	12	8	7	6	12	1	6	2	6	6		2	2		2	2						126
16	20		2	4	6	2	5	4	18	4	4	2	2	8		3	2	2		4								92
24	2	4	7	6		8			1		8		2	4	2	4		8	2	2			4					64
32	5			3			1		5					2														16
36						1							2															3
48	3			1	2	5	2		1		6			2		4				2	2			2				32
64	2			4					4											1		2			1			14
72	2												2															4
80	1																											1
96	3								1							1												5
108													2															2
120						1																						1
144	2												6											2				10
192									1							1												2
216	1					1																						2
256	1																											1
288	1																											1
576	1																											1
720																1												1
768	1																											1
960																									1			1
1296																										2		2
1536									1																			1
1728																1												1
2592																								2				2
4320																1										1		2
11 520	1																											1
2 280 960																											1	1
All	187	292	187	332	424	353	334	592	557	668	696	368	290	446	276	154	142	88	34	16	35	6	4	6	2	3	1	6493

all optimal codes of length 11 and 12 required 80 and 320 days of CPU time, respectively, and a parallel cluster computer was used for this search. We observed that most of this time was spent on computing minimum distance to eliminate non-optimal codes, and much less time on canonizing the optimal codes.

Although this paper has focused on codes over $\mathrm{GF}(9)$, our classification algorithm can be generalized to Hermitian self-dual additive codes over $\mathrm{GF}(q = m^2)$ for any prime power m. (One simply needs to find an appropriate coordinate graph, as discussed in Section III.) The results in this paper also has applications beyond the study of additive codes. The correspondence between self-dual additive codes over $\mathrm{GF}(9)$ and 3-weighted graphs means that we have also classified particular classes of 3-weighted graphs that should be of interest in graph theory. An equivalence class of self-dual additive codes over $\mathrm{GF}(9)$ maps to an orbit of graphs under generalized local complementation [4], [22]. Orbits of graphs with respect to local complementation has a long history in combinatorics [20], [28], with several applications, for instance in the theory of *interlace polynomials* [29], [30]. The generalization to weighted graphs is a natural next step. The results in this paper also have applications in the field of quantum information theory. Our previous classification of codes over $\mathrm{GF}(4)$ [12] has since led to new results in the study of the *entanglement* of *quantum graph states* [31], and the new data obtained in this paper will yield similar insights into the properties of ternary quantum graph states.

ACKNOWLEDGEMENT

Thanks to Parallab, the High Performance Computing unit of the University of Bergen, whose cluster computer made the computational results in this paper possible. The author would like to thank the anonymous reviewers for providing useful suggestions and corrections that improved the quality of the manuscript.

REFERENCES

[1] G. Nebe, E. M. Rains, and N. J. A. Sloane, *Self-dual codes and invariant theory*, ser. Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 2006, vol. 17.

[2] E. M. Rains, "Nonbinary quantum codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1827–1832, 1999.

[3] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1369–1387, 1998.

[4] L. E. Danielsen, "Graph-based classification of self-dual additive codes over finite fields," *Adv. Math. Commun.*, vol. 3, no. 4, pp. 329–348, 2009.

[5] G. Höhn, "Self-dual codes over the Kleinian four group," *Math. Ann.*, vol. 327, no. 2, pp. 227–255, 2003.

[6] M. Van den Nest, "Local equivalence of stabilizer states and codes," Ph.D. dissertation, K. U. Leuven, Belgium, May 2005.

[7] D. G. Glynn, T. A. Gulliver, J. G. Maks, and M. K. Gupta, "The geometry of additive quantum codes," 2006, preprint. [Online]. Available: http://www.maths.adelaide.edu.au/rey.casse/DavidGlynn/QMonoDraft.pdf

[8] T. A. Gulliver, J.-L. Kim, and Y. Lee, "New MDS or near-MDS self-dual codes," *IEEE Trans. Inform. Theory*, vol. 54, no. 9, pp. 4354–4360, 2008.

[9] J.-L. Kim and Y. Lee, "Euclidean and Hermitian self-dual MDS codes over large finite fields," *J. Combin. Theory Ser. A*, vol. 105, no. 1, pp. 79–95, 2004.

[10] M. Grassl and T. A. Gulliver, "On circulant self-dual codes over small fields," *Des. Codes Cryptogr.*, vol. 52, no. 1, pp. 57–81, 2009.

[11] P. Gaborit, "Quadratic double circulant codes over fields," *J. Combin. Theory Ser. A*, vol. 97, no. 1, pp. 85–107, 2002.

[12] L. E. Danielsen and M. G. Parker, "On the classification of all self-dual additive codes over GF(4) of length up to 12," *J. Combin. Theory Ser. A*, vol. 113, no. 7, pp. 1351–1367, 2006.

[13] J. H. Conway, V. Pless, and N. J. A. Sloane, "Self-dual codes over GF(3) and GF(4) of length not exceeding 16," *IEEE Trans. Inform. Theory*, vol. 25, no. 3, pp. 312–322, 1979.

[14] M. Harada, C. Lam, A. Munemasa, and V. D. Tonchev, "Classification of generalized Hadamard matrices H(6,3) and quaternary Hermitian self-dual codes of length 18," *Electron. J. Combin.*, vol. 17, no. 1, 2010.

[15] M. Harada and A. Munemasa, "Classification of quaternary Hermitian self-dual codes of length 20," *IEEE Trans. Inform. Theory*, vol. 57, no. 6, pp. 3758–3762, 2011.

[16] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli, "Nonbinary stabilizer codes over finite fields," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4892–4914, 2006.

[17] P. R. J. Östergård, "Classifying subspaces of Hamming spaces," *Des. Codes Cryptogr.*, vol. 27, no. 3, pp. 297–305, 2002.

[18] D. Schlingemann, "Stabilizer codes can be realized as graph codes," *Quantum Inf. Comput.*, vol. 2, no. 4, pp. 307–323, 2002.

[19] M. Grassl, A. Klappenecker, and M. Rötteler, "Graphs, quadratic forms, and quantum codes," in *Proc. IEEE Int. Symp. Inform. Theory*, 2002, p. 45.

[20] A. Bouchet, "Graphic presentations of isotropic systems," *J. Combin. Theory Ser. B*, vol. 45, no. 1, pp. 58–76, 1988.

[21] M. Van den Nest, J. Dehaene, and B. De Moor, "Graphical description of the action of local Clifford transformations on graph states," *Phys. Rev. A*, vol. 69, no. 2, p. 022316, 2004.

[22] M. Bahramgiri and S. Beigi, "Graph states under the action of local Clifford group in non-binary case," 2006, preprint. [Online]. Available: http://arxiv.org/abs/quant-ph/0610267

[23] M. Harada and P. R. J. Östergård, "On the classification of self-dual codes over $\mathbb{F}_5$," *Graphs Combin.*, vol. 19, no. 2, pp. 203–214, 2003.

[24] ——, "Self-dual and maximal self-orthogonal codes over $\mathbb{F}_7$," *Discrete Math.*, vol. 256, no. 1-2, pp. 471–477, 2002.

[25] B. D. McKay, *nauty User's Guide, Version 2.4*, Nov. 2009. [Online]. Available: http://cs.anu.edu.au/~bdm/nauty/nug.pdf

[26] P. Gaborit, W. C. Huffman, J.-L. Kim, and V. Pless, "On additive GF(4) codes," in *Codes and Association Schemes*, ser. DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 56. Providence, RI: Amer. Math. Soc., 2001, pp. 135–149.

[27] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*. San Diego, CA: Academic Press, 1995.

[28] H. de Fraysseix, "Local complementation and interlacement graphs," *Discrete Math.*, vol. 33, no. 1, pp. 29–35, 1981.

[29] M. Aigner and H. van der Holst, "Interlace polynomials," *Linear Algebra Appl.*, vol. 377, pp. 11–30, 2004.

[30] L. E. Danielsen and M. G. Parker, "Interlace polynomials: Enumeration, unimodality, and connections to codes," *Discrete Appl. Math.*, vol. 158, no. 6, pp. 636–648, 2010.

[31] A. Cabello, L. E. Danielsen, A. J. López-Tarrida, and J. R. Portillo, "Optimal preparation of graph states," *Phys. Rev. A*, vol. 83, no. 4, p. 042314, 2011.