Improved Semidefinite Programming Bound on Sizes of Codes

Hyun Kwang Kim, Member, IEEE, and Phan Thanh Toan

Abstract

Let $A(n, d)$ (respectively $A(n, d, w)$) be the maximum possible number of codewords in a binary code (respectively binary constant-weight $w$ code) of length $n$ and minimum Hamming distance at least $d$. By adding new linear constraints to Schrijver’s semidefinite programming bound, which is obtained from block-diagonalising the Terwilliger algebra of the Hamming cube, we obtain two new upper bounds on $A(n, d)$, namely $A(18, 8) \leq 71$ and $A(19, 8) \leq 131$. Twenty three new upper bounds on $A(n, d, w)$ for $n \leq 28$ are also obtained by a similar way.

Index Terms

Binary codes, binary constant-weight codes, linear programming, semidefinite programming, upper bound.

I. INTRODUCTION

Let $F = \{0, 1\}$ and let $n$ be a positive integer. The (Hamming) distance between two vectors in $F^n$ is the number of coordinates where they differ. The (Hamming) weight of a vector in $F^n$ is the distance between it and the zero vector. The minimum distance of a subset of $F^n$ is the smallest distance between any two different vectors in that subset. An $(n, d)$ code is a subset of $F^n$ having minimum distance $\geq d$. If $C$ is an $(n, d)$ code, then an element of $C$ is called a codeword and the number of codewords in $C$ is called the size of $C$.

The largest possible size of an $(n, d)$ code is denoted by $A(n, d)$. The problem of determining the exact values of $A(n, d)$ is one of the most fundamental problems in combinatorial coding theory. Among upper bounds on $A(n, d)$, Delsarte’s linear programming bound is quite powerful (see $[1]$ and $[2]$). This bound is obtained from block-diagonalising the Bose-Mesner algebra of $F^n$. In 2005, by block-diagonalising the Terwilliger algebra (which contains the Bose-Mesner algebra) of $F^n$, Schrijver gave a semidefinite programming bound $[3]$. This bound was shown to be stronger than or as good as Delsarte’s linear programming bound. In fact, eleven new upper bounds on $A(n, d)$ were obtained in the paper for $n \leq 28$. In 2002, Mounits, Etzion, and Litsyn added more linear constraints to Delsarte’s linear programming bound and obtained new upper bounds on $A(n, d)$ $[4]$. In this paper, we construct new linear constraints and show that these linear constraints improve Schrijver’s semidefinite programming bound.

The authors are with the Department of Mathematics, Pohang University of Science and Technology, Pohang 790-784, Korea (e-mail: hkkim@postech.ac.kr; pttoan@postech.ac.kr).
Among improved upper bounds on \( A(n, d) \) for \( n \leq 28 \), there are two new upper bounds, namely \( A(18, 8) \leq 71 \) and \( A(19, 8) \leq 131 \).

An \((n, d, w)\) constant-weight code is an \((n, d)\) code such that every codeword has weight \(w\). Let \( A(n, d, w) \) be the largest possible size of an \((n, d, w)\) constant-weight code. The problem of determining the exact values of \( A(n, d, w) \) has its own interest. Upper bounds on \( A(n, d, w) \) can even help to improve upper bounds on \( A(n, d) \) (for example, see [4], [2]). There are also Delsarte’s linear programming bound and Schrijver’s semidefinite programming bound on \( A(n, d, w) \) [1], [3]. In 2000, Agrell, Vardy, and Zeger added new linear constraints to Delsarte’s linear programming bound and improved several upper bounds on \( A(n, d, w) \) [5]. More linear constraints that improve upper bounds on \( A(n, d, w) \) can be found in [6]. In this paper, we add further new linear constraints to Schrijver’s semidefinite programming bound on \( A(n, d, w) \) and obtain twenty three new upper bounds on \( A(n, d, w) \) for \( n \leq 28 \).

II. Upper Bounds on \( A(n, d) \)

In this section, we improve upper bounds on \( A(n, d) \) by adding more linear constraints to Schrijver’s semidefinite programming bound, which is obtained from block-diagonalising the Terwilliger algebra of the Hamming cube \( F^n \).

For more details about Schrijver’s semidefinite programming bound, see [3].

A. General Definition of \( A(n, d) \) and \( A(n, d, w) \)

We first give a general definition. Let \( n \) and \( d \) be positive integers. For a finite (possibly empty) set \( \Lambda = \{(X_i, d_i)\}_{i \in I} \), where each \( X_i \) is a vector in \( F^n \) and each \( d_i \) is a nonnegative integer, we define

\[
A(n, \Lambda, d) = \text{maximum possible number of codewords in a binary code of length } n \text{ and minimum distance } \geq d \text{ such that each codeword is at distance } d_i \text{ from } X_i, \forall i \in I. \tag{1}
\]

1) \( |\Lambda| = 0 \): If \( \Lambda \) is empty, then we get the usual definition of \( A(n, d) \).

2) \( |\Lambda| = 1 \): If \( \Lambda \) contains only one element, says \((X_1, d_1)\), then \( A(n, \Lambda, d) \) is the maximum possible number of codewords in a binary code of length \( n \) and minimum distance \( \geq d \) such that each codeword is at distance \( d_1 \) from \( X_1 \). By translation, we may assume that \( X_1 \) is the zero vector so that each codeword has weight \( d_1 \). Therefore,

\[
A(n, \Lambda, d) = A(n, d, w), \tag{2}
\]

where \( w = d_1 \).

A \((w_1, n_1, w_2, n_2, d)\) doubly-constant-weight code is an \((n_1 + n_2, d, w_1 + w_2)\) constant-weight code such that every codeword has exactly \(w_1\) ones on the first \(n_1\) coordinates (and hence has exactly \(w_2\) ones on the last \(n_2\) coordinates). Let \( T(w_1, n_1, w_2, n_2, d) \) be the largest possible size of a \((w_1, n_1, w_2, n_2, d)\) doubly-constant-weight
code. Agrell, Vardy, and Zeger showed in [5] that upper bounds on \( T(w_1, n_1, w_2, n_2, d) \) can help improving upper bounds on \( A(n, d, w) \). In our result, upper bounds on \( T(w_1, n_1, w_2, n_2, d) \) will be used to improve upper bounds on \( A(n, d) \). As \( A(n, d) \) and \( A(n, d, w) \), \( T(w_1, n_1, w_2, n_2, d) \) is also a special case of \( A(n, \Lambda, d) \).

3) \(|\Lambda| = 2\): If \( \Lambda \) contains two elements, then the following proposition shows that \( A(n, \Lambda, d) \) is exactly \( T(w_1, n_1, w_2, n_2, d) \).

**Proposition 1:** If \( \Lambda = \{(X_1, d_1), (X_2, d_2)\} \), then

\[
A(n, \Lambda, d) = T(w_1, n_1, w_2, n_2, d),
\]

where \( n_1 = d(X_1, X_2), n_2 = n - n_1, w_1 = \frac{1}{2}(d_1 - d_2 + n_1), \) and \( w_2 = \frac{1}{2}(d_1 + d_2 - n_1) \).

**Proof:** Let \( n_1 = d(X_1, X_2) \) and \( n_2 = n - n_1 \). By translation, we may assume that \( X_1 \) is the zero vector. Hence, \( d(X_1, X_2) = wt(X_2) \). Let \( Y \) be a vector at distance \( d_1 \) from \( X_1 \) and at distance \( d_2 \) from \( X_2 \). By rearranging the coordinates, we may assume that

\[
\begin{align*}
X_1 & = \begin{array}{cccccccccccc}
n_1 & & & & & & & & & & & 0 \cdots 0 0 \cdots 0 & \end{array} \\
X_2 & = \begin{array}{cccccccccccc}
1 & \cdots & 1 & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
Y & = & 0 & \cdots & 0 & 1 & \cdots & 1 & 1 & \cdots & 0 \end{array}.
\end{align*}
\]

Since \( X_1 \) is the zero vector, we have

\[
w_1 + w_2 = wt(Y) = d(Y, X_1) = d_1.
\]

Also,

\[
(n_1 - w_1) + w_2 = d(Y, X_2) = d_2.
\]

\( \square \)

\( \square \) and (5) give \( w_1 = \frac{1}{2}(d_1 - d_2 + n_1) \) and \( w_2 = \frac{1}{2}(d_1 + d_2 - n_1) \).

4) \(|\Lambda| \geq 3\): It becomes more complicated when \( \Lambda \) contains more than two elements. We consider a very special case when \( |\Lambda| = 4 \), which will be used in our improving upper bounds on \( A(n, d, w) \) in Section III. Suppose that \( \Lambda = \{(X_1, d_1), (X_2, d_2), (X_3, d_3), (X_4, d_4)\} \) satisfies the following conditions.

- \( X_1 \) is the zero vector (which can always be assumed).
- \( X_2 \) and \( X_3 \) have the same weight \( d_1 \).
- \( X_4 = X_2 + X_3 \).

Then \( A(n, \Lambda, d) = T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \), where \( w_i \) and \( n_i \) \((1 \leq i \leq 4)\) are determined in the next proposition. The definition of \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \) is similar to that of \( T(n_1, w_1, n_2, w_2, d) \) (it is the largest possible size of a \((\sum_{i=1}^{4} n_i, d)\) code such that on each codeword there are exactly \( w_i \) ones on the \( n_i \) coordinates \((1 \leq i \leq 4)\)).

**Proposition 2:** Suppose that \( \Lambda = \{(X_i, d_i)\}_{i=1}^{4} \) satisfies \( X_1 \) is the zero vector, \( wt(X_2) = wt(X_3) = d_1 \), and \( X_4 = X_2 + X_3 \). Then

\[
A(n, \Lambda, d) = T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d),
\]
where \( n_1 = n_3 = \frac{1}{2} d(X_2, X_3), n_2 = d_1 - n_1, n_4 = n - n_1 - n_2 - n_3, \)

\[
\begin{align*}
    w_1 &= \frac{1}{4} (d_1 - d_2 + d_3 - d_4) + \frac{1}{2} n_1, \\
    w_2 &= \frac{1}{4} (d_1 - d_2 - d_3 + d_4) + \frac{1}{2} n_2, \\
    w_3 &= \frac{1}{4} (d_1 + d_2 - d_3 - d_4) + \frac{1}{2} n_3, \\
    w_4 &= \frac{1}{4} (d_1 + d_2 + d_3 + d_4) + \frac{1}{2} (n_4 - n).
\end{align*}
\]

**Proof:** Suppose that \( Z \) is a vector at distance \( d_i \) from \( X_i \) (\( 1 \leq i \leq 4 \)). By rearranging the coordinates, we may assume the following.

\[
\begin{align*}
    X_2 &= \overbrace{0 \ldots 0}^{n_2} 1 \overbrace{0 \ldots 0}^{n_4}, \\
    X_3 &= 0 \overbrace{0 \ldots 0}^{n_3} 1 \ldots 1, \\
    Z &= 0 \overbrace{\ldots 0}^{w_1} 0 \overbrace{\ldots 0}^{w_2} 1 \ldots 1 0 \overbrace{\ldots 0}^{w_4}.
\end{align*}
\]

Let \( n_1, n_2, n_3, n_4 \) be as in the above figure. Since \( n_1 + n_3 = d(X_2, X_3) \) and \( X_2, X_3 \) have the same weight, \( n_1 = n_3 = \frac{1}{2} d(X_2, X_3) \). Now \( n_1 + n_2 = wt(X_2) = d_1 \). Therefore, \( n_2 = d_1 - n_1 \) and \( n_4 = n - n_1 - n_2 - n_3 \). We have

\[
\begin{align*}
    w_1 + w_2 + w_3 + w_4 &= wt(Z) = d(Z, X_1) = d_1, \\
    (n_1 - w_1) + (n_2 - w_2) + w_3 + w_4 &= d(Z, X_2) = d_2, \\
    w_1 + (n_2 - w_2) + (n_3 - w_3) + w_4 &= d(Z, X_3) = d_3, \\
    (n_1 - w_1) + w_2 + (n_3 - w_3) + w_4 &= d(Z, X_4) = d_4.
\end{align*}
\]

Solving these equations, we get \( w_i \) (\( 1 \leq i \leq 4 \)) as desired.

**B. Schrijver’s Semidefinite Programming Bound on \( A(n, d) \)**

Let \( \mathcal{P} \) be the collection of all subsets of \( \{1, 2, \ldots, n\} \). Each vector in \( \mathcal{F}^n \) can be identified with its support (the support of a vector is the set of coordinates at which the vector has nonzero entries). With this identification, a code is a subset of \( \mathcal{P} \) and the (Hamming) distance between two subsets \( X \) and \( Y \) in \( \mathcal{P} \) is \( d(X, Y) = |X \Delta Y| \). Let \( \mathcal{C} \) be an \( (n, d) \) code. For each \( i, j \), and \( t \), define

\[
x_{i,j}^t = \frac{1}{|\mathcal{C}|} \frac{1}{n \choose i-t,j-t,t} \lambda_{i,j}^t,
\]

where \( \binom{a,b,c}{b_1,b_2,\ldots,b_m} \) denotes the number of pairwise disjoint subsets of sizes \( b_1, b_2, \ldots, b_m \) respectively of a set of size \( a \), and \( \lambda_{i,j}^t \) denotes the number of triples \( (X, Y, Z) \in \mathcal{C} \^3 \) with \( |X \Delta Y| = i, |X \Delta Z| = j, \) and \( |(X \Delta Y) \cap (X \Delta Z)| = t \), or equivalently, with \( |X \Delta Y| = i, |X \Delta Z| = j, \) and \( |Y \Delta Z| = i+j-2t \). Set \( x_{i,j}^t = 0 \) if \( \binom{n}{i-t,j-t,t} = 0 \).
The key part of Schrijver’s semidefinite programming bound is that for each $k = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, the matrices
\[
\left( \sum_{t=0}^{n} \beta_{i,j,k}^t x_{i,j}^t \right)_{i,j=k}^{n-k}
\]
and
\[
\left( \sum_{t=0}^{n} \beta_{i,j,k}^t (x_{i+j-2t,0}^t - x_{i,j}^t) \right)_{i,j=k}^{n-k}
\]
are positive semidefinite, where $\beta_{i,j,k}^t$ is given by
\[
\beta_{i,j,k}^t = \sum_{u=0}^{n} (-1)^{u-t} {u \choose t} \left( \frac{n - 2k}{u - k} \right) \left( \frac{n - k - u}{i - u} \right) \left( \frac{n - k - u}{j - u} \right).
\]
Since
\[
|C| = \sum_{i=0}^{n} {n \choose i} x_{i,0}^0,
\]
an upper bound on $A(n,d)$ can be obtained by considering the $x_{i,j}^t$ as variables and by
\[
\text{maximizing } \sum_{i=0}^{n} {n \choose i} x_{i,0}^0
\]
subject to the matrices (8) and (9) are positive semidefinite for each $k = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$ and subject to the following conditions on the $x_{i,j}^t$ (see (3)).

(i) $x_{0,0}^0 = 1$.
(ii) $0 \leq x_{i,j}^t \leq x_{i,0}^0$ and $x_{i,0}^0 + x_{j,0}^0 \leq 1 + x_{i,j}^t$ for all $i, j, t \in \{0, 1, \ldots, n\}$.
(iii) $x_{i,j}^t = x_{i',j'}^t$ if $(i', j', i+j-2t)$ is a permutation of $(i, j, i+j-2t)$.
(iv) $x_{i,j}^t = 0$ if $\{i, j, i+j-2t\} \cap \{1, 2, \ldots, d-1\} \neq \emptyset$.

C. Improved Schrijver’s Semidefinite Programming Bound on $A(n,d)$

1) New Constraints for $x_{i,j}^t$: Let $C$ be an $(n,d)$ code and let $x_{i,j}^t$ be defined by (7).

Theorem 3: For all $i, j, t \in \{0, 1, \ldots, n\}$ with \( \binom{n}{i-t,j-t} \) \neq 0,
\[
x_{i,j}^t \leq T(t,i,j-t,n-i,d) \binom{n-i}{j-t} x_{i,0}^0.
\]

Proof: Recall that $\lambda_{i,j}^t$ is the number of triples $(X,Y,Z) \in C^3$ with $|X \Delta Y| = i$, $|X \Delta Z| = j$, and $|Y \Delta Z| = i + j - 2t$. For any pair $(X,Y) \in C^2$ with $|X \Delta Y| = i$, the number of $Z \in C$ such that $|Z \Delta X| = j$ and $|Z \Delta Y| = i + j - 2t$ is upper bounded by $A(n,\Lambda, d)$, where $\Lambda = \{(X, j), (Y, i+j-2t)\}$. By Proposition \[1\]
\[
A(n,\Lambda, d) = T(t,i,j-t,n-i,d).
\]
Since the number of pairs $(X,Y) \in C^2$ such that $|X \Delta Y| = i$ is $\lambda_{i,0}^0$,
\[
\lambda_{i,j}^t \leq T(t,i,j-t,n-i,d) \lambda_{i,0}^0.
\]
Therefore,

\[ x^t_{i,j} = \frac{1}{|C|} \binom{n}{i-t,j-t,t} \lambda^t_{i,j} \]

\[ \leq \frac{T(t,i,j-t,n-i,d)}{|C|} \binom{n}{i-t,j-t,t} \lambda^0_{i,0} \]

\[ = \frac{T(t,i,j-t,n-i,d)}{\binom{n}{i-t,j-t,t}} \binom{n}{n-i} \binom{n-j}{j-t} \]

\[ \leq T(t,i,j-t,n-i,d) \binom{n}{i-t,j-t,t} \lambda^0_{i,0} \]

\[ = T(t,i,j-t,n-i,d) \binom{n-i}{n-j} \binom{n-j}{j-t} \]

\[ = T(t,i,j-t,n-i,d) x^0_{i,0}. \]

The following corollary was used in [3].

**Corollary 4:** For each \( j \in \{0, 1, \ldots, n\} \),

\[ \binom{n}{j} x^0_{0,j} \leq A(n, d, j). \] (16)

**Proof:** By Theorem 3, we have

\[ x^0_{0,j} \leq T(0, 0, j, n, d) x^0_{0,0} = A(n, d, j). \] (17)

**Remark 5:** Theorem 3 improve the condition \( x^t_{i,j} \leq x^0_{i,0} \) in Schrijver’s semidefinite programming bound since \( \frac{T(t,i,j-t,n-i,d)}{\binom{n-i}{n-j} \binom{n-j}{j-t}} \leq 1 \) (in fact, \( \frac{T(t,i,j-t,n-i,d)}{\binom{n-i}{n-j} \binom{n-j}{j-t}} \) is much less than 1 in general). Similarly, Corollary 4 in many cases (of \( i \) and \( j \)) improve the condition \( x^0_{i,0} + x^0_{j,0} \leq 1 + x^t_{i,j} \) since \( x^0_{0,0} = x^0_{0,0} = A(n, d, u) \binom{n-u}{n} \) is much less than \( \frac{1}{2} \) in general.

2) Delsarte’s Linear Programming Bound and Its Improvements: Let \( C \) be an \((n,d)\) code, the distance distribution \( \{B_i\}_{i=0}^n \) of \( C \) is defined by

\[ B_i = \frac{1}{|C|} \cdot |\{(X,Y) \in C^2 \mid |X \Delta Y| = i\}|. \] (18)

By definition,

\[ \binom{n}{i} x^0_{i,0} = B_i \] (19)

for each \( i = 0, 1, \ldots, n \). Hence, \( \{\binom{n}{i} x^0_{i,0}\}_{i=0}^n \) is the distance distribution on \( C \). The following result can be found for example in [7] or [6].

**Theorem 6:** (Delsarte’s linear programming bound and its improvements). Let \( C \) be an \((n,d)\) code with distance distribution \( \{B_i\}_{i=0}^n \). For \( k = 1, 2, \ldots, n \),

\[ \sum_{i=1}^{n} P_k(n;i) B_i \geq -\binom{n}{k}, \] (20)
where \( P_k(n; x) \) is the Krawtchouk polynomial given by
\[
P_k(n; x) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j}.
\] (21)

If \( M = |C| \) is odd, then
\[
\sum_{i=1}^{n} P_k(n; i)B_i \geq -\binom{n}{k} + \frac{1}{M} \binom{n}{k}.
\] (22)

If \( M = |C| \equiv 2 \pmod{4} \), then there exists \( t \in \{0, 1, \ldots, n\} \) such that
\[
\sum_{i=1}^{n} P_k(n; i)B_i \geq -\binom{n}{k} + 2M \left[ \binom{n}{k} + P_k(n; t) \right].
\] (23)

3) Linear Constraints on Distance Distributions \( \{B_i\}_{i=0}^{n} \): If some linear constraints are used to improve Del- sarte’s linear programming bound on \( A(n, d) \), then these constraints can still be added to Schrijver’s semidefinite programming bound to improve upper bounds on \( A(n, d) \). The following constraints are due to Mounits, Etzion, and Litsyn (see [4, Theorems 9 and 10]).

**Theorem 7:** Let \( C \) be an \( (n, d) \) code with distance distribution \( \{B_i\}_{i=0}^{n} \). Suppose that \( d \) is even and \( \delta = d/2 \). Then
\[
B_{n-\delta} + \left\lfloor \frac{n}{\delta} \right\rfloor \sum_{i<\delta} B_{n-i} \leq \left\lfloor \frac{n}{\delta} \right\rfloor
\] (24)
and
\[
B_{n-\delta-i} + [A(n, d, \delta+i) - A(n-\delta+i, d, \delta+i)]B_{n-\delta+i} + A(n, d, \delta+i) \sum_{j>i} B_{n-\delta+j} \leq A(n, d, \delta+i)
\] (25)
for all \( i = 1, 2, \ldots, \delta - 1 \).

Table I shows improved upper bounds on \( A(n, d) \) when linear constraints in Theorems 3, 6, and 7 are added to Schrijver’s semidefinite programming bound (12). In the table, by Schrijver bound we mean upper bound obtained from Schrijver’s semidefinite programming bound (12). Among improved upper bounds on \( A(n, d) \), there are two new upper bounds, namely
\[
A(18, 8) \leq 71 \quad \text{and} \quad A(19, 8) \leq 131.
\]

The other best known upper bounds are from [8]. As in [3], all computations here were done by the algorithm SDPT3 available online on the NEOS Server for Optimization (http://www.neos-server.org/neos/solvers/index.html).

**Remark 8:** Since \( A(n, d) = A(n + 1, d + 1) \) if \( d \) is odd, we can always assume that \( d \) is even. If \( d \) is even, then \( A(n, d) \) is attained by a code with all codewords having even weights. Hence, in Schrijver’s semidefinite programming bound, one can put \( x_{i,j}^t = 0 \) if \( i \) or \( j \) is odd.

**Remark 9:** In Theorems 3 and 7, the values of \( A(n, d, w) \) and \( T(w_1, n_1, w_2, n_2, d) \) may have not yet been known. However, we can replace them by any of their upper bounds (see the proof of [4] Theorem 10) for the validity of this replacement in Theorem 7. While best known upper bounds on \( A(n, d, w) \) (which are mostly from [9], [5], [3], [10]) are used in our computations, all upper bounds on \( T(w_1, n_1, w_2, n_2, d) \) that we used are from the tables on Erik Agrell’s website (http://webfiles.portal.chalmers.se/s2/research/kit/bounds/dcw.html).
TABLE I
IMPROVED UPPER BOUNDS FOR A(n, d)

<table>
<thead>
<tr>
<th>n</th>
<th>d</th>
<th>best lower bound</th>
<th>best upper bound previously known</th>
<th>new upper bound Schrijver</th>
<th>improved bound Schrijver</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>8</td>
<td>64</td>
<td>72</td>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>128</td>
<td>135</td>
<td>131</td>
<td>142</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>256</td>
<td>256</td>
<td>262</td>
<td>274</td>
</tr>
<tr>
<td>25</td>
<td>8</td>
<td>4096</td>
<td>5421</td>
<td>5465</td>
<td>5477</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>4104</td>
<td>9275</td>
<td>9649</td>
<td>9697</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>384</td>
<td>836</td>
<td>885</td>
<td>886</td>
</tr>
<tr>
<td>25</td>
<td>12</td>
<td>52</td>
<td>55</td>
<td>57</td>
<td>58</td>
</tr>
<tr>
<td>26</td>
<td>12</td>
<td>64</td>
<td>96</td>
<td>97</td>
<td>98</td>
</tr>
</tbody>
</table>

III. UPPER BOUNDS ON A(n, d, w)

A. Some Properties of A(n, d, w)

We begin with some elementary properties of A(n, d, w) which can be found in [2].

Theorem 10:

\[ A(n, d, w) = A(n, d + 1, w), \quad \text{if } d \text{ is odd,} \quad (26) \]

\[ A(n, d, w) = A(n, d, n - w), \quad (27) \]

\[ A(n, 2, w) = \binom{n}{w}, \quad (28) \]

\[ A(n, 2w, w) = \left\lfloor \frac{n}{w} \right\rfloor, \quad (29) \]

\[ A(n, d, w) = 1, \quad \text{if } 2w < d. \quad (30) \]

Remark 11: By (26) and (28), we can always assume that d is even and \( d \geq 4 \). Also, by (27), (29), and (30), we can assume that \( d < 2w \leq n \).

B. Schrijver’s Semidefinite Programming Bound on A(n, d, w)

Let \( C \) be an (n, d, w) constant-weight code and let \( v = n - w \). For each \( t, s, i, \) and \( j \), define

\[ y_{t,s}^{i,j} = \frac{1}{|C|} \left( \binom{w}{i-t, j-t, t} \binom{w}{i-s, j-s, s} \right)^{1/2}, \quad (31) \]
where \( \mu_{i,j}^{t,s} \) is the number of triples \((X,Y,Z) \in C^3\) with \(|X \setminus Y| = i, |X \setminus Z| = j, (X \setminus Y) \cap (X \setminus Z)| = t,\) and \(|(Y \setminus X) \cap (Z \setminus X)| = s,\) or equivalently, with \(|X\Delta Y| = 2i, |X\Delta Z| = 2j, |Y\Delta Z| = 2(i + j - t - s),\) and \(|X\Delta Y\Delta Z| = w + 2t - 2s.\) Set \( y_{i,j}^{t,s} = 0 \) if either \( (i-t-w-t,t) \) or \( (v-v-s, j-s-s,s) \) is 0.

In the previous section, \( \beta_{i,j,k}^t \) depends on \( n.\) Hence, \( \beta_{i,j,k}^t \) should be denoted by \( \beta_{i,j,k}^{t,n}.\) We will use the later notation in this section. As in [3], for each \( k = 0, 1, \ldots, \lfloor \frac{n}{w} \rfloor \) and each \( l = 0, 1, \ldots, \lfloor \frac{n}{w} \rfloor,\) the matrices

\[
\left( \sum_{i,j=1}^{n} \frac{\beta_{i,j,k}^{t,w} \beta_{i,j,l}^{s,v} \beta_{i,j,l}^{s,v} y_{i,j}^{t,s}}{i,j \in W_k \cap V_l} \right)_{i,j \in W_k \cap V_l}
\]

and

\[
\left( \sum_{i,j=1}^{n} \frac{\beta_{i,j,k}^{t,w} \beta_{i,j,l}^{s,v} \beta_{i,j,l}^{s,v} (y_{i+j-s-0}^{t,s} - y_{i,j}^{t,s})}{i,j \in W_k \cap V_l} \right)_{i,j \in W_k \cap V_l}
\]

are positive semidefinite, where \( W_k \) \( \{k, k+1, \ldots, w-k\} \) and \( V_l \) \( \{l, l+1, \ldots, v-l\}.\) Since

\[
|C| = \sum_{i=0}^{\min\{w,v\}} \binom{w}{i} \binom{v}{i} y_{i,0}^{0,0},
\]

an upper bound on \( A(n,d,w) \) can be obtained by considering the \( y_{i,j}^{t,s} \) as variables and by

\[
\text{maximizing} \sum_{i=0}^{\min\{w,v\}} \binom{w}{i} \binom{v}{i} y_{i,0}^{0,0}
\]

subject to the matrices \( \text{32} \) and \( \text{33} \) are positive semidefinite for each \( k = 0, 1, \ldots, \lfloor \frac{n}{w} \rfloor \) and each \( l = 0, 1, \ldots, \lfloor \frac{n}{w} \rfloor,\) and subject to the following conditions.

(i) \( y_{0,0}^{0,0} = 1.\)

(ii) \( 0 \leq y_{i,j}^{t,s} \leq y_{i,0}^{0,0} \) and \( y_{i,0}^{0,0} + y_{i,j}^{0,0} \leq 1 + y_{i,j}^{t,s} \) for all \( i, j, t, s \in \{0, 1, \ldots, \min\{w, v\}\}.\)

(iii) \( y_{i,j}^{t,s} = y_{i',j'}^{t',s'} \) if \( t' - s' = t - s \) and \( i', j', i' + j' - t' - s' \) is a permutation of \( (i, j, i + j - t - s).\)

(iv) \( y_{i,j}^{t,s} = 0 \) if \( \{2i, 2j, 2(i + j - t)\} \cap \{1, 2, \ldots, d-1\} \neq \emptyset.\)

C. Improved Schrijver’s Semidefinite Programming Bound on \( A(n,d,w)\)

1) New Constraints for \( y_{i,j}^{t,s}:\) Let \( C \) be an \( (n, d, w) \) constant-weight code and let \( y_{i,j}^{t,s} \) be defined by \( \text{31}.\) The following theorem corresponds to Theorem \( \text{3} \) in the previous section.

**Theorem 12:** For all \( i, j, t, s \in \{0, 1, \ldots, \min\{w, v\}\} \) with \( \left( \frac{w}{i-t-j-t} \right) \neq 0 \) and \( \left( \frac{v}{i-s-j-s-s} \right) \neq 0,\)

\[
y_{i,j}^{t,s} \leq T(t,i,j-t,w-i,s,i,j-s,v-i,d) y_{i,0}^{0,0}.
\]

**Proof:** Suppose that \((X,Y) \in C^2\) such that \(|X\Delta Y| = 2i.\) We claim that the number of codewords \( Z \in C\) such that \(|X\Delta Z| = 2i, |Y\Delta Z| = 2(i + j - t - s),\) and \(|X\Delta Y\Delta Z| = w + 2t - 2s \) is upper bounded by \( T(t,i,j-t,w-i,s,i,j-s,v-i,d).\) It is easy to see that this number is upper bounded by \( A(n, \Lambda, d),\) where \( \Lambda = \{(0,w), (X,2j), (Y,2(i+j-t-s)), (X,Y,w+2t-2s)\}.\) By Proposition \( \text{2}\)

\[
A(n, \Lambda, d) = T(w_1,n_1,w_2,n_2,w_3,n_3,w_4,n_4,d),
\]

(37)
where \( n_1 = n_3 = \frac{1}{2} |X \Delta Y| = i \), \( n_2 = d_1 - n_1 = w - i \), \( n_4 = n - i - (w - i) - i = v - i \), and similarly, \( w_1 = i - t \), \( w_2 = (w - i) - (j - t) \), \( w_3 = s \), \( w_4 = j - s \). Hence,

\[
A(n, A, d) = T(i - t, i, (w - i) - (j - t), w - i, s, i, j - s, v - i, d)
\]

\[
= T(t, i, j - t, w - i, s, i, j - s, v - i, d),
\]

(38)

where the later equality comes from Proposition 22 (iii) in the appendix. Since the number of pairs \((X, Y) \in C^2\) such that \(|X \Delta Y| = 2i\) is \(\mu_{i,0}^{0,0}\),

\[
\mu_{i,j}^{t,s} \leq T(t, i, j - t, w - i, s, i, j - s, v - i, d)\mu_{i,0}^{0,0}.
\]

(39)

Therefore,

\[
y_{i,j}^{t,s} = \frac{1}{|C|} \left( \begin{array}{c} w \\ i \\ t \\ j \end{array} \right) \left( \begin{array}{c} v \\ s \\ j \end{array} \right) \mu_{i,j}^{t,s} \leq \frac{T(t, i, j - t, w - i, s, i, j - s, v - i, d)\mu_{i,0}^{0,0}}{|C|} \left( \begin{array}{c} w \\ i \\ t \\ j \end{array} \right) \left( \begin{array}{c} v \\ s \\ j \end{array} \right) y_{i,0}^{0,0}.
\]

\[
y_{i,j}^{t,s} \leq T(t, i, j - t, w - i, s, i, j - s, v - i, d)\mu_{i,0}^{0,0}
\]

(40)

2) Delsarte’s Linear Programming Bound: Let \( C \) be an \((n, d, w)\) constant-weight code with distance distribution \(\{B_{i,j}\}_{i=0}^{n}\). By definition of \(y_{i,j}^{t,s}\),

\[
\left( \begin{array}{c} w \\ i \\ t \\ j \end{array} \right) \left( \begin{array}{c} v \\ s \\ j \end{array} \right) y_{i,0}^{0,0} = B_{2i}
\]

(41)

for every \( i \) (note that \( B_0 = 1 \) and \( B_i = 0 \) whenever \( i \) is odd or \( 0 < i < d \) or \( i > 2w \)).

Theorem 13: (Delsarte’s linear programming bound). If \(\{B_{i,j}\}_{i=0}^{n}\) is the distance distribution of an \((n, d, w)\) constant-weight code, then for \( k = 1, 2, \ldots, w \),

\[
\sum_{i=d/2}^{w} q(k, i, n, w)B_{2i} \geq -1,
\]

(42)

where

\[
q(k, i, n, w) = \frac{\sum_{j=0}^{i} (-1)^j \binom{k}{j} \binom{w-k}{i-j} \binom{n-w-k}{i-j}}{\binom{w}{i} \binom{n-w}{i}}.
\]

Specifying Delsarte’s linear programming bound on \( A(n, d) \) gives the following linear constraints on \( B_{i,j} \), which sometimes help reducing upper bounds on \( A(n, d, w) \) by 1 (see [6] Proposition 11).

Theorem 14: Let \( C \) be an \((n, d, w)\) constant-weight code with distance distribution \(\{B_{i,j}\}_{i=0}^{n}\). For each \( k = 1, 2, \ldots, n \),

\[
\sum_{i=d/2}^{w} P_k\{(n; 2i)B_{2i} \leq \frac{2}{M} \left[ \binom{n}{k} - r_k \right] q_k(M - q_k) + r_k(q_k + 1)(M - q_k - 1) \},
\]

(43)
where $q_k$ and $r_k$ are the quotient and the remainder, respectively, when dividing $MP_k^-(n; w)$ by $\binom{n}{k}$, i.e.

$$MP_k^-(n; w) = q_k \binom{n}{k} + r_k$$

with $0 \leq r_k < \binom{n}{k}$, and where $P_k^-(n; x)$ is defined by

$$P_k^-(n; x) = \sum_{j=0}^{n} \binom{x}{j} \binom{n-x}{k-j}.$$  \hfill (45)

3) New Linear Constraints on Distance Distributions $\{B_i\}_{i=0}^{n}$. Linear constraints which correspond to those in Theorem 7 have not been studied for constant-weight codes even though similar constraints have been studied by Argrell, Vardy, and Zeger in \cite{Argrell} (see Theorem 21 below). We now present these constraints. Several new notations are needed. For convenience, we fix the following settings until the end of this section.

- $C$ is an $(n, d, w)$ constant-weight code with distance distribution $\{B_i\}_{i=0}^{n}$ such that $d$ is even and $d < 2w \leq n$.
- Let $v = n - w$. Since $2w \leq n$, $w \leq v$.
- Let $H = \{d/2, d/2 + 1, \ldots, w\}$, which is the set of all positive integer $i$ such that $B_{2i}$ can be nonzero.
- For each $i \in H$, let $V_i$ be the set of all vectors $X$ in $F^n$ such that $X$ has exactly $i$ ones on the first $w$ coordinates and exactly $i$ ones on the last $v = n - w$ coordinates.
- For $i \neq j$ both in $H$, define

$$m_{i,j} = \max\{d(X, Y) \mid X \in V_i, Y \in V_j\}. \hfill (46)$$

- For each codeword $X$ in $C$, let

$$S_{2i}(X) = \{Y \in C \mid d(X, Y) = 2i\}, \hfill (47)$$

which is the set of all codewords $Y$ in $C$ at distance $2i$ from $X$. By definition of $\{B_i\}_{i=0}^{n}$,

$$B_{2i} = \frac{1}{|C|} \sum_{X \in C} |S_{2i}(X)|$$

for each $i \in H$.
- For each $i \in H$, let $Q_i$ denote an integer such that

$$T(i, w, i, v, d) \leq Q_i.$$ \hfill (49)

- For $i \neq j$ both in $H$ with $i + j \geq v$ and $m_{i,j} = d$, let $Q_{ji}$ denote an integer such that

$$T(w - j, i, v - j, i, d) \leq Q_{ji}. \hfill (50)$$

**Proposition 15:** For $i \neq j$ both in $H$,

$$m_{i,j} = a + b,$$ \hfill (51)

where

$$a = \begin{cases} 
  i + j & \text{if } i + j < w \\
  i + j - 2(i + j - w) & \text{if } i + j \geq w
\end{cases}$$
and

\[ b = \begin{cases} 
  i + j & \text{if } i + j < v \\
  i + j - 2(i + j - v) & \text{if } i + j \geq v 
\end{cases} \]

In particular, if \( i + j \geq v \geq w \), then

\[ m_{i,j} = 2(n - i - j). \]  

(52)

**Proof:** The proof is straightforward.

\[ \square \]

Lemma 16: For each \( i \in H \) and each codeword \( X \in \mathcal{C} \),

\[ |S_{2i}(X)| \leq Q_i. \]  

(53)

**Proof:** Let \( X \) be a codeword in \( \mathcal{C} \). It is easy to see that \( |S_{2i}(X)| \) is upper bounded by \( A(n, \Lambda, d) \), where \( \Lambda = \{(0, w), (X, 2i)\} \). By Propositions \[1\] and \[2\] (iii),

\[ A(n, \Lambda, d) \leq T(w - i, w, i, v, d) = T(i, w, i, v, d). \]  

(54)

Hence, \( |S_{2i}(X)| \leq T(i, w, i, v, d) \leq Q_i. \)

\[ \square \]

Theorem 17: Suppose that \( H_1 \) is a nonempty subset of \( H \) such that \( m_{i,j} < d \) for all \( i \neq j \) both in \( H_1 \). Then for each codeword \( X \in \mathcal{C} \), \( S_{2i}(X) \) is nonempty for at most one \( i \) in \( H_1 \). Furthermore,

\[ \sum_{i \in H_1} B_{2i} \leq 1. \]  

(55)

**Proof:** Let \( X \) be a codeword in \( \mathcal{C} \). Suppose on the contrary that there exist \( i \neq j \) both in \( H_1 \) such that \( S_{2i}(X) \) and \( S_{2j}(X) \) are nonempty. Then choose any \( Y \in S_{2i}(X) \) and \( Z \in S_{2j}(X) \). By rearranging the coordinates, we may assume that

\[ X = \begin{array}{c} w \\
 i \ldots 1 \\
 v \ldots 0 \end{array} \]  

(56)

Since \( d(X,Y) = 2i \) and \( X \) and \( Y \) have the same weight \( w \), \( Y + X \) must have exactly \( i \) ones on the first \( w \) coordinates and exactly \( i \) ones on the last \( v \) coordinates. This means \( Y + X \in \mathcal{V}_i \). Similarly, \( Z + X \in \mathcal{V}_j \). By definition of \( m_{i,j} \), \( d(Y + X, Z + X) \leq m_{i,j} \). Thus,

\[ d(Y, Z) = d(Y + X, Z + X) \leq m_{i,j} < d, \]  

(57)

which is a contradiction since \( Y \) and \( Z \) are two different codewords in \( \mathcal{C} \). Hence, \( S_{2i}(X) \) is nonempty for at most one \( i \) in \( H_1 \). It follows by Lemma \[16\] that

\[ \sum_{i \in H_1} \frac{|S_{2i}(X)|}{Q_i} \leq 1. \]  

(58)

Taking sum of (58) over all \( X \in \mathcal{C} \), we get

\[ \sum_{i \in H_1} \frac{B_{2i}}{Q_i} \leq 1. \]  

(59)

\[ \square \]
We now consider the case $m_{i,j} = d$ for some $i \neq j$ both in $H$. The following Lemma says that the existence of a codeword at distance $2i$ from $X$ may reduce the total number of codewords at distance $2j$ from $X$.

**Lemma 18:** Suppose $i \neq j$ both in $H$ such that $i + j \geq v$ and $m_{i,j} = d$. If $X$ is a codeword in $C$ such that $|S_{2i}(X)| \geq 1$, then

$$|S_{2j}(X)| \leq Q_{ji}.$$  \hfill (60)

**Proof:** Fix a codeword $Y \in S_{2i}(X)$. If $S_{2j}(X)$ is empty, then there is nothing to prove. Hence, we assume $|S_{2j}(X)| \geq 1$. Let $Z \in S_{2j}(X)$. By rearranging the coordinates, we may assume that

$$X = \begin{bmatrix} w & \cdots & 0 \end{bmatrix}$$  \hfill (61)

As in the proof of Theorem 17, we can show that $Y + X \in V_i$ and $Z + X \in V_j$. By definition of $m_{i,j}$,

$$d \leq d(Y, Z) = d(Y + X, Z + X) \leq m_{i,j} = d.$$  \hfill (62)

Thus,

$$d(Y, Z) = d(Y + X, Z + X) = m_{i,j} = d.$$  \hfill (63)

Since $i + j \geq v \geq w$, by rearranging the first $w$ coordinates, we may assume that on the first $w$ coordinates:

$$Y + X = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} w-i \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \cdots$$
$$Z + X = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} w-j \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \cdots$$  \hfill (64)

On the first $w$ coordinates, $Z + X$ must have exactly $i + j - w$ ones on the first $i$ coordinates (the other $w - i$ ones of $Z + X$ must be fixed since $d(Y + X, Z + X) = m_{i,j}$).

Similarly, since $i + j \geq v$, by rearranging the last $v$ coordinates, we may assume that on the last $v$ coordinates:

$$Y + X = \begin{bmatrix} \cdots & 1 \cdots \end{bmatrix} \begin{bmatrix} v-i \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}$$
$$Z + X = \begin{bmatrix} \cdots & 0 \cdots \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} v-j \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \cdots$$  \hfill (65)

On the last $v$ coordinates, $Z + X$ must have exactly $i + j - v$ ones on the first $i$ coordinates (the other $v - i$ ones of $Z + X$ must be fixed since $d(Y + X, Z + X) = m_{i,j}$).

From (61), (64), and (65), we get

$$d(Z, X + Y) = wt(X + Y + Z)$$
$$= wt(X + (Y + X) + (Z + X))$$
$$= (i + j - w) + (v - j + v - i)$$
$$= 2v - w.$$  \hfill (66)
Now the number of $Z \in S_{2j}(X)$ is upper bounded by \( A(n, \Lambda, d) \), where \( \Lambda = \{(0, w), (X, 2j), (Y, d), (X + Y, 2v - w)\} \). By Proposition [15]

\[
d = m_{i,j} = 2(n - i - j).
\]

(67)

Applying Proposition [2] we get (by replacing \( d = 2(n - i - j) \) and \( n = w + v \))

\[
A(n, \Lambda, d) = T(w - j, i, 0, w - i, i + j - v, i, v - i, v - i, d)
= T(w - j, i, v - j, i, d).
\]

(68)


where the last equality comes from Proposition [22] in the appendix. Therefore,

\[
|S_{2j}(X)| \leq A(n, \Lambda, d) = T(w - j, i, v - j, i, d) \leq Q_{ji}.
\]

(69)

**Theorem 19:** Suppose that \( H_1 \) is a subset of \( H \) satisfying the following properties.

- \( |H_1| \geq 2 \).
- There exist \( i \neq j \) both in \( H_1 \) such that \( i + j \geq v \) and \( m_{i,j} = d \).
- For all \( k \neq l \) both in \( H_1 \) such that either \( k \notin \{i, j\} \) or \( l \notin \{i, j\} \), we always have \( m_{k,l} < d \).

Let \( H_2 = H_1 \setminus \{i, j\} \). Then

\[
\frac{Q_j - Q_{ji}}{Q_j Q_{ij}} B_{2i} + \frac{1}{Q_j} B_{2j} + \sum_{k \in H_2} \frac{1}{Q_k} B_{2k} \leq 1, \quad \text{if} \quad \frac{Q_{ij}}{Q_i} + \frac{Q_{ji}}{Q_j} \geq 1,
\]

(70)

\[
\frac{1}{Q_i} B_{2i} + \frac{Q_i - Q_{ij}}{Q_i Q_{ji}} B_{2j} + \sum_{k \in H_2} \frac{1}{Q_k} B_{2k} \leq 1, \quad \text{if} \quad \frac{Q_{ij}}{Q_i} + \frac{Q_{ji}}{Q_j} \geq 1,
\]

(71)

\[
\sum_{k \in H_1} \frac{1}{Q_k} B_{2k} \leq 1, \quad \text{if} \quad \frac{Q_{ij}}{Q_i} + \frac{Q_{ji}}{Q_j} \leq 1.
\]

(72)

**Proof:** We first prove (70). It suffices to show that for every codeword \( X \) in \( C \),

\[
\frac{Q_j - Q_{ji}}{Q_j Q_{ij}} |S_{2i}(X)| + \frac{1}{Q_j} |S_{2j}(X)| + \sum_{k \in H_2} \frac{1}{Q_k} |S_{2k}(X)| \leq 1,
\]

if \( \frac{Q_{ij}}{Q_i} + \frac{Q_{ji}}{Q_j} \geq 1 \). Let \( X \) be any codeword in \( C \). By Lemma [19]

\[
|S_{2i}(X)| \leq Q_i \quad \text{and} \quad |S_{2j}(X)| \leq Q_j.
\]

(74)

By Lemma [18]

\[
|S_{2i}(X)| \leq Q_{ij} \quad \text{if} \quad |S_{2j}(X)| \geq 1,
\]

(75)

\[
|S_{2j}(X)| \leq Q_{ji} \quad \text{if} \quad |S_{2i}(X)| \geq 1.
\]

(76)

We prove (73) by considering the following three cases.
Case 1: \( |S_{2i}(X)| = 0 \). Proving (73) is exactly the same as proving (58). So we are done.

Case 2: \( |S_{2i}(X)| \geq 1 \) and \( |S_{2j}(X)| = 0 \). Since \( |S_{2i}(X)| \geq 1 \) and \( |S_{2k}(X)| = 0 \) for every \( k \in H_2 \) by Theorem 17

Hence, to prove (73), we only need to prove that

\[
(Q_j - Q_{ji})|S_{2i}(X)| \leq Q_j Q_{ij}.
\]

(77)

By hypothesis, \( \frac{Q_{ji}}{Q_j} + \frac{Q_{ij}}{Q_{ij}} \geq 1 \). Thus, \( (Q_j - Q_{ji})Q_i \leq Q_j Q_{ij} \) and hence

\[
(Q_j - Q_{ji})|S_{2i}(X)| \leq (Q_j - Q_{ji})Q_i \leq Q_j Q_{ij}.
\]

(78)

\[\begin{align*}
\text{Case 3: } |S_{2i}(c)| &\geq 1 \text{ and } |S_{2j}(c)| \geq 1. \text{ As in Case 2, } |S_{2k}(X)| = 0 \text{ for every } k \in H_2. \text{ We have} \\
\frac{Q_j - Q_{ij}}{Q_j Q_{ij}}|S_{2i}(X)| + \frac{1}{Q_j}|S_{2j}(X)| &\leq \frac{Q_j - Q_{ij}}{Q_j Q_{ij}}Q_{ij} + \frac{1}{Q_j}Q_{ij} \\
&= 1 - \frac{Q_{ij}}{Q_j} + \frac{Q_{ij}}{Q_j} \\
&= 1.
\end{align*}\]

(79)

Therefore, (73) is proved and so is (70).

By symmetry, (71) follows.

We now prove (72). It suffices to show that for every codeword \( X \) in \( C \),

\[
\sum_{k \in H_1} \frac{1}{Q_k}|S_{2k}(X)| \leq 1,
\]

(80)

if \( \frac{Q_{ji}}{Q_j} + \frac{Q_{ij}}{Q_{ij}} \leq 1 \). If either \( |S_{2i}(X)| = 0 \) or \( |S_{2j}(X)| = 0 \), then proving (80) is exactly the same as proving (58). Hence, suppose that \( |S_{2i}(X)| \geq 1 \) and \( |S_{2j}(X)| \geq 1 \). As in Case 2, \( |S_{2k}(X)| = 0 \) for every \( k \in H_2 \). We have

\[
\frac{1}{Q_i}|S_{2i}(X)| + \frac{1}{Q_j}|S_{2j}(X)| \leq \frac{1}{Q_i}Q_{ij} + \frac{1}{Q_j}Q_{ij} \leq 1.
\]

(81)

\[\square\]

We now specify which \( H_1 \) are used in Theorems 17 and 19. Let

\[
\alpha = d/2 - (n - 2w)
\]

(82)

and let

\[
\alpha_1 = \left\lfloor \frac{\alpha + 1}{2} \right\rfloor \text{ and } \alpha_2 = \left\lceil \frac{\alpha}{2} \right\rceil
\]

(83)

so that \( \alpha_1 + \alpha_2 = \alpha \). Also, let

\[
i_0 = w - \alpha_1 \text{ and } j_0 = w - \alpha_2.
\]

(84)

- Case 1: \( \alpha \) is even. In this case, \( i_0 = j_0 \). We apply Theorem 17 for

\[
H_1 = \{j_0, j_0 + 1, \ldots, w\}
\]

(85)

and apply Theorem 19 for

\[
H_1 = \{i_0 - \epsilon, j_0 + \epsilon, j_0 + \epsilon + 1, \ldots, w\}
\]

(86)
(with \(i = i_0 - \epsilon\) and \(j = j_0 + \epsilon\) for each \(\epsilon = 1, 2, \ldots, w - j_0\).

- **Case 2: \(\alpha\) is odd.** In this case, \(i_0 < j_0\). We apply Theorem 19 for

\[
H_1 = \{i_0 - \epsilon, j_0 + \epsilon, j_0 + \epsilon + 1, \ldots, w\}
\]

(with \(i = i_0 - \epsilon\) and \(j = j_0 + \epsilon\) for each \(\epsilon = 0, 1, \ldots, w - j_0\).

**Example 20:** Consider \((n, d, w) = (27, 8, 13)\). We have \(\alpha = d/2 - (n - 2w) = 3\) is odd. Hence, \(\alpha_1 = 2\) and \(\alpha_2 = 1\). So, \(i_0 = 11\) and \(j_0 = 12\). We can apply Theorem 19 for \(H_1 = \{i = i_0, j = j_0, w\} = \{11, 12, 13\}\) (with \(\epsilon = 0\)). We have

\[
Q_i = 26 \geq T(2, 13, 3, 14, 8) = T(11, 13, 11, 14, 8),
Q_j = 1 = T(1, 13, 2, 14, 8) = T(12, 13, 12, 14, 8),
Q_{ij} = 20 \geq T(2, 12, 3, 12, 8),
Q_{ji} = 1 = T(1, 11, 2, 11, 8),
\]

and

\[
Q_k = 1 = T(0, 13, 1, 14, 8) = T(13, 13, 13, 14, 8)
\]

for \(k = 13\). Since \(\frac{Q_{ij}}{Q_i} + \frac{Q_{ji}}{Q_j} = \frac{20}{26} + \frac{1}{1} \geq 1\), Theorem 19 gives

\[
B_{24} + B_{26} \leq 1
\]

and

\[
\frac{1}{26} B_{22} + \frac{26 - 20}{26} B_{24} + B_{26} \leq 1.
\]

The later constraint is equivalent to

\[
B_{22} + 6B_{24} + 26B_{26} \leq 26.
\]

For \(H_1 = \{10, 13\}\) (with \(\epsilon = 1\)), Theorem 19 gives less effective linear constraints.

When \(\alpha \leq 0\), there is no set \(H_1\) satisfying Theorem 19. In this case, the following type of linear constraints which comes from [5, Proposition 17] is very useful. As in [5], let \(T'(w_1, n_1, w_2, n_2, d)\) be the largest possible size of a \((w_1, n_1, w_2, n_2, d)\) doubly-bounded-weight code (a \((w_1, n_1, w_2, n_2, d)\) doubly-bounded-weight code is an \((n_1 + n_2, d, w_1 + w_2)\) constant-weight code such that every codeword has at most \(w_1\) ones on the first \(n_1\) coordinates). Tables for upper bounds on \(T'(w_1, n_1, w_2, n_2, d)\) can be found on Erik Agrell’s website [http://webfiles.portal.chalmers.se/s2/research/kit/bounds/dbw.html](http://webfiles.portal.chalmers.se/s2/research/kit/bounds/dbw.html).

**Theorem 21:** Let \(\delta = d/2\). For \(i, j \in \{\delta, \delta + 1, \ldots, w\}\) with \(i \neq j\). If \(i + j \leq n - \delta\), define \(P_{ij}\) and \(P_{ji}\) as any nonnegative integers such that

\[
P_{ij} \geq \min\{P_i, T'(\Delta, i - \Delta, n - w - j, 2i - 2\Delta)\},
\]

\[
P_{ji} \geq \min\{P_j, T'(\Delta, i - \Delta, n - w - i, 2j - 2\Delta)\}.
\]
TABLE II
NEW UPPER BOUNDS FOR $A(n, d, w)$

<table>
<thead>
<tr>
<th>n</th>
<th>d</th>
<th>w</th>
<th>best lower bound</th>
<th>best upper bound previously known</th>
<th>new upper bound</th>
<th>Schrijver bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>6</td>
<td>8</td>
<td>588</td>
<td>1107</td>
<td>1106</td>
<td>1136</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>10</td>
<td>616</td>
<td>734</td>
<td>630</td>
<td>634</td>
</tr>
<tr>
<td>23</td>
<td>8</td>
<td>9</td>
<td>400</td>
<td>707</td>
<td>703</td>
<td>707</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>9</td>
<td>887</td>
<td>2108</td>
<td>2104</td>
<td>2108</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>11</td>
<td>1988</td>
<td>5225</td>
<td>5208</td>
<td>5225</td>
</tr>
<tr>
<td>27</td>
<td>8</td>
<td>9</td>
<td>1023</td>
<td>2914</td>
<td>2882</td>
<td>2918</td>
</tr>
<tr>
<td>27</td>
<td>8</td>
<td>11</td>
<td>2404</td>
<td>7833</td>
<td>7754</td>
<td>7833</td>
</tr>
<tr>
<td>27</td>
<td>8</td>
<td>12</td>
<td>3335</td>
<td>10547</td>
<td>10460</td>
<td>10697</td>
</tr>
<tr>
<td>27</td>
<td>8</td>
<td>13</td>
<td>4094</td>
<td>11981</td>
<td>11897</td>
<td>11981</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>9</td>
<td>1333</td>
<td>3895</td>
<td>3886</td>
<td>3900</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>11</td>
<td>3773</td>
<td>11939</td>
<td>11896</td>
<td>12025</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>12</td>
<td>4927</td>
<td>17011</td>
<td>17008</td>
<td>17011</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>13</td>
<td>6848</td>
<td>21152</td>
<td>21148</td>
<td>21152</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>9</td>
<td>45</td>
<td>81</td>
<td>79</td>
<td>82</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>11</td>
<td>125</td>
<td>380</td>
<td>379</td>
<td>380</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>12</td>
<td>137</td>
<td>434</td>
<td>433</td>
<td>434</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>11</td>
<td>168</td>
<td>566</td>
<td>565</td>
<td>566</td>
</tr>
<tr>
<td>26</td>
<td>10</td>
<td>12</td>
<td>208</td>
<td>702</td>
<td>691</td>
<td>702</td>
</tr>
<tr>
<td>27</td>
<td>10</td>
<td>11</td>
<td>243</td>
<td>882</td>
<td>871</td>
<td>882</td>
</tr>
<tr>
<td>27</td>
<td>10</td>
<td>12</td>
<td>351</td>
<td>1201</td>
<td>1190</td>
<td>1201</td>
</tr>
<tr>
<td>27</td>
<td>10</td>
<td>13</td>
<td>405</td>
<td>1419</td>
<td>1406</td>
<td>1419</td>
</tr>
<tr>
<td>28</td>
<td>10</td>
<td>11</td>
<td>308</td>
<td>1356</td>
<td>1351</td>
<td>1356</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>12</td>
<td>28</td>
<td>37</td>
<td>36</td>
<td>37</td>
</tr>
</tbody>
</table>

where $\Delta := w - \delta$. Also, define $P_k := Q_k$ for each $k \in H$. Then

$$P_{ji}B_{2i} + (P_i - P_{ij})B_{2j} \leq P_iP_{ji}, \quad \text{if } \frac{P_{ij}}{P_i} + \frac{P_{ji}}{P_j} > 1,$$  \hspace{1cm} (93)

$$P_jB_{2i} + P_iB_{2j} \leq P_jP_i, \quad \text{if } \frac{P_{ij}}{P_i} + \frac{P_{ji}}{P_j} \leq 1.$$  \hspace{1cm} (95)
By adding the linear constraints in Theorems [12] [14] [17] [19] and [21] to Schrijver’s semidefinite programming bound [35], we obtained new upper bounds on \( A(n, d, w) \) shown on Table II. As before, all computations were done by the same algorithm SDPT3 at the same server.

**APPENDIX**

**UPPER BOUNDS ON \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \)**

To apply Theorem [12] we need tables of upper bounds on \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \). However, there are no such tables available since this is the first time the function \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \) is introduced. We show here some elementary properties that are used to obtain upper bounds on \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \).

In general, let us define \( T(\{(w_i, n_i)\}^t_{i=1}, d) \) as follows. For \( t \geq 1 \), a \( \{(w_i, n_i)\}^t_{i=1}, d \) multiply constant-weight code is a \( \{(\sum^t_{i=1} n_i, d) \) code such that there are exactly \( w_i \) ones on the \( n_i \) coordinates. When \( t = 1 \) this is definition of an \( (n_1, d_1, w_1) \) constant-weight code, when \( t = 2 \) this is definition of a \( (w_1, n_1, w_2, n_2, d_2) \) doubly-constant-weight code, etc. Let \( T(\{(w_i, n_i)\}^t_{i=1}, d) \) be the largest possible size of a \( \{(w_i, n_i)\}^t_{i=1}, d \) multiply constant-weight code.

We present here elementary properties that are used to get upper bounds on \( T(\{(w_i, n_i)\}^t_{i=1}, d) \). The proofs of these properties are similar to those for \( A(n, d, w) \) or \( T(w_1, n_1, w_2, n_2, d) \), and hence are omitted. Upper bounds on \( T(w_1, n_1, w_2, n_2, w_3, n_3, w_4, n_4, d) \) that we used in Theorem [12] are the best upper bounds obtained from these properties.

**Proposition 22:** (i) If \( d \) is odd then,

\[
T(\{(w_i, n_i)\}^t_{i=1}, d) = T(\{(w_i, n_i)\}^t_{i=1}, d + 1).
\]

(ii) If \( w_j = 0 \) for some \( j \in \{1, 2, \ldots , t\} \), then

\[
T(\{(w_i, n_i)\}^t_{i=1}, d) = T(\{(w_i, n_i)\}_{i \neq j}, d).
\]

(iii) \( T(\{(w_i, n_i)\}^t_{i=1}, d) \) does not change if we replace any \( w_i \) by \( n_i - w_i \).

(iv) \( T(\{(w_i, n_i)\}^t_{i=1}, 2) = \prod^t_{i=1} \binom{n_i}{w_i} \).

(v) \( T(\{(w_i, n_i)\}^t_{i=1}, 2 \sum^t_{i=1} w_i) = \min_{1 \leq i \leq t} \left\lfloor \frac{n_i}{w_i} \right\rfloor \).

(vi) \( T(\{(w_i, n_i)\}^t_{i=1}, d) = 1 \) if \( 2 \sum^t_{i=1} w_i < d \).

**Remark 23:** By (i) and (iv), we can always assume that \( d \) is even and \( d \geq 4 \). By (ii) and (iii), we may assume that \( 0 < 2w_i \leq n_i \) for each \( i \). Also, by (v) and (vi), we can assume that \( d \leq 2 \sum^t_{i=1} w_i \).

The next proposition can be used to reduce the size of \( \{(w_i, n_i)\}^t_{i=1} \) from \( t \) to \( t - 1 \). When the size of the set is 2, we use known upper bounds on \( T(w_1, n_1, w_2, n_2, d) \).

**Proposition 24:** If \( t \geq 2 \), then

\[
T(\{(w_i, n_i)\}^t_{i=1}, d) \leq T(\{(w_i', n_i')\}^{t-1}_{i=1}, d),
\]

where \( w'_i = w_i, n'_i = n_i \) for \( i = 1, 2, \ldots , t - 2 \), and \( w'_{t-1} = w_{t-1} + w_t, n'_{t-1} = n_{t-1} + n_t \).

**Proposition 25:** If \( w_i > 0 \), then

\[
T(\{(w_i, n_i)\}^t_{i=1}, d) \leq \left\lfloor \frac{n_i}{w_i} T(\{(w'_i, n'_i)\}^t_{i=1}, d) \right\rfloor,
\]

where \( w'_i = w_i, n'_i = n_i \) for \( i = 1, 2, \ldots , t - 2 \), and \( w'_{t-1} = w_{t-1} + w_t, n'_{t-1} = n_{t-1} + n_t \).
where \( \{(w', n')\}_{i=1}^{t} \) is obtained from \( \{(w_i, n_i)\}_{i=1}^{t} \) by replacing the pair \((w_i, n_i)\) by \((w_i - 1, n_i - 1)\).

**Proposition 26:** If \( w_i < n_i \), then

\[
T(\{(w_i, n_i)\}_{i=1}^{t}, d) \leq \left\lfloor \frac{n_i}{n_i - w_i} T(\{(w'_i, n'_i)\}_{i=1}^{t}, d) \right\rfloor,
\]

where \( \{(w'_i, n'_i)\}_{i=1}^{t} \) is obtained from \( \{(w_i, n_i)\}_{i=1}^{t} \) by replacing the pair \((w_i, n_i)\) by \((w_i - 1, n_i - 1)\).

**References**


