Minimal lenghts for codes with given primal and dual distance

Iliya Bouyukliev[1] Erik Jacobsson[2]

[1]Institute of Mathematics and Informatics
Bulgarian Academy of Sciences

[2]Department of Mathematical Sciences
University of Gothenburg/Chalmers University of Technology

Sixth International Workshop on Optimal Codes and Related Topics: OC 2009

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Contents

Motivation and background

In cryptography, in order to obscure the relationship between the ciphertext and the key, substitution boxes (S-boxes) are generally used to transform S input bits into T output bits.

An S-box is a collection of T Boolean functions $f : GF(2)^S \rightarrow GF(2)$.

The security of a block cipher against various attacks comes down to the security of the S-Boxes, which in turn comes down to the security of the Boolean functions.

Motivation and background

In cryptography, in order to obscure the relationship between the ciphertext and the key, substitution boxes (S-boxes) are generally used to transform S input bits into T output bits.

An S-box is a collection of T Boolean functions $f : GF(2)^S \rightarrow GF(2)$.

The security of a block cipher against various attacks comes down to the security of the S-Boxes, which in turn comes down to the security of the Boolean functions.

Motivation and background

In cryptography, in order to obscure the relationship between the ciphertext and the key, substitution boxes (S-boxes) are generally used to transform S input bits into T output bits.

An S-box is a collection of T Boolean functions $f : GF(2)^S \longrightarrow GF(2)$.

The security of a block cipher against various attacks comes down to the security of the S-Boxes, which in turn comes down to the security of the Boolean functions.

Motivation and background

Definition

A Boolean function $f : GF(2)^S \rightarrow GF(2)$ is called K-resilient if we can fix any set of K, $K < S$, input bits and the function gives 0 and 1 equally often, on the remaining 2^{S-K} different inputs.

Definition

A Boolean function $f : GF(2)^S \rightarrow GF(2)$ is said to satisfy propagation criteria, $PC(L)$ if for a fixed $x \in GF(2)^S$

$$f(x) - f(x + \Delta)$$

gives 0 and 1 equally often, for $\Delta \in GF(2)^S$ with Hamming weight $1 \leq w(\Delta) \leq L$

Motivation and background

Definition

A Boolean function $f : GF(2)^S \to GF(2)$ is called K-resilient if we can fix any set of K, $K < S$, input bits and the function gives 0 and 1 equally often, on the remaining 2^{S-K} different inputs.

Definition

A Boolean function $f : GF(2)^S \to GF(2)$ is said to satisfy propagation criteria, $PC(L)$ if for a fixed $x \in GF(2)^S$

$$f(x) - f(x + \Delta)$$

gives 0 and 1 equally often, for $\Delta \in GF(2)^S$ with Hamming weight $1 \leq w(\Delta) \leq L$

Motivation and background

Definition

A Boolean function $f : GF(2)^S \rightarrow GF(2)$ *is said to satisfy the extended propagation criteria,* $EPC(L)$ *of order* K *if*

$$f(x) - f(x + \Delta)$$

is K*-resilient for* $\Delta \in GF(2)^S$ *with* $1 \leq w(\Delta) \leq L$.

In fact, it has been shown that the $EPC(L)$ of order K is directly related to security of a Boolean function against both linear and differential attacks.

Motivation and background

Question:

Given L and K, what is the minimum S for which an $EPC(L)$ of order K function exists?

Theorem (Kurosawa and Satoh(1997))

There exists an $EPC(L)$ function $f(x_1, ..., x_S)$ of order K if there exists a linear code of length $\frac{S}{2}$, some dimension, minimum distance $K+1$ and dual distance $L+1$.

If we let $n = \frac{S}{2}$, $d = K+1$, $d^{\perp} = L+1$ and let k denote the dimension we can reformulate the question.

Motivation and background

Question:

Given L and K, what is the minimum S for which an $EPC(L)$ of order K function exists?

Theorem (Kurosawa and Satoh(1997))

There exists an $EPC(L)$ function $f(x_1, ..., x_S)$ of order K if there exists a linear code of length $\frac{S}{2}$, some dimension, minimum distance $K+1$ and dual distance $L+1$.

If we let $n = \frac{S}{2}$, $d = K+1$, $d^{\perp} = L+1$ and let k denote the dimension we can reformulate the question.

Question:

Given L and K, what is the minimum S for which an $EPC(L)$ of order K function exists?

Theorem (Kurosawa and Satoh(1997))

There exists an $EPC(L)$ function $f(x_1, ..., x_S)$ of order K if there exists a linear code of length $\frac{S}{2}$, some dimension, minimum distance $K + 1$ and dual distance $L + 1$.

If we let $n = \frac{S}{2}$, $d = K + 1$, $d^{\perp} = L + 1$ and let k denote the dimension we can reformulate the question.

Definitions and Notations

Reformulated question:

What is the least n such that there exists a linear code of length n with minimum distance d and dual distance $d^{\perp}$, where d and $d^{\perp}$ are fixed?

Definition (Matsumoto et.al. 2004)

$N(d, d^{\perp}) =$ *The minimum n such that there exists a linear $[n, k, d]$ code with dual distance $d^{\perp}$.*

Definitions and Notations

Reformulated question:

What is the least n such that there exists a linear code of length n with minimum distance d and dual distance $d^\perp$, where d and $d^\perp$ are fixed?

Definition (Matsumoto et.al. 2004)

$N(d, d^\perp) =$ *The minimum n such that there exists a linear $[n, k, d]$ code with dual distance $d^\perp$.*

Objectives

- Find some values for $N(d, d^{\perp})$ for specific d and $d^{\perp}$.
- For these values classify all inequivalent codes reaching $N(d, d^{\perp})$.

Objectives

- Find some values for $N(d, d^{\perp})$ for specific d and $d^{\perp}$.
- For these values classify all inequivalent codes reaching $N(d, d^{\perp})$.

History of the problem

1. The problem to study the function $N(d, d^{\perp})$ was given by Matsumoto et al. in 2006.
   They presented:
   - Some general bounds on the function $N(d, d^{\perp})$
     (I.e. new versions of known bounds Griesmer, Hamming, linear programming bound).
   - Some examples (although no systematical investigation of the exact values of $N(d, d^{\perp})$).
2. Kohnert gave a talk in 2008 on construction of linear codes having prescribed primal-dual minimum distances. The construction gave new upper bounds on $N(d, d^{\perp})$.

History of the problem

1. The problem to study the function $N(d, d^{\perp})$ was given by Matsumoto et al. in 2006.
   They presented:
   - Some general bounds on the function $N(d, d^{\perp})$
     (I.e. new versions of known bounds Griesmer, Hamming, linear programming bound).
   - Some examples (although no systematical investigation of the exact values of $N(d, d^{\perp})$).
2. Kohnert gave a talk in 2008 on construction of linear codes having prescribed primal-dual minimum distances. The construction gave new upper bounds on $N(d, d^{\perp})$.

History of the problem

1. The problem to study the function $N(d, d^{\perp})$ was given by Matsumoto et al. in 2006.
   They presented:
   - Some general bounds on the function $N(d, d^{\perp})$ (I.e. new versions of known bounds Griesmer, Hamming, linear programming bound).
   - Some examples (although no systematical investigation of the exact values of $N(d, d^{\perp})$).
2. Kohnert gave a talk in 2008 on construction of linear codes having prescribed primal-dual minimum distances. The construction gave new upper bounds on $N(d, d^{\perp})$.

History of the problem

1. The problem to study the function $N(d, d^{\perp})$ was given by Matsumoto et al. in 2006.
   They presented:
   - Some general bounds on the function $N(d, d^{\perp})$
     (I.e. new versions of known bounds Griesmer, Hamming, linear programming bound).
   - Some examples (although no systematical investigation of the exact values of $N(d, d^{\perp})$).
2. Kohnert gave a talk in 2008 on construction of linear codes having prescribed primal-dual minimum distances. The construction gave new upper bounds on $N(d, d^{\perp})$.

Preliminaries

Theorem

Let C be a linear code with minimum distance d and dual distance $d^{\perp}$, and let C' be the punctured code of C. Then C' has minimum distance at least $d-1$ and dual distance at least $d^{\perp}$.

For $d, d^{\perp} > 2$, we have

$$N(d-1, d^{\perp}) \leq N(d, d^{\perp}) - 1$$

$$N(d, d^{\perp}-1) \leq N(d, d^{\perp}) - 1.$$

I.e. the $N(d, d^{\perp})$ function is strictly increasing in both its arguments.

Preliminaries

Theorem

Let C be a linear code with minimum distance d and dual distance $d^\perp$, and let C' be the punctured code of C. Then C' has minimum distance at least $d-1$ and dual distance at least $d^\perp$.

For $d, d^\perp > 2$, we have

$$N(d-1, d^\perp) \leq N(d, d^\perp) - 1$$

$$N(d, d^\perp - 1) \leq N(d, d^\perp) - 1.$$

I.e. the $N(d, d^\perp)$ function is strictly increasing in both its arguments.

Definition

Let G be a generator matrix of a linear binary $[n, k, d]$ code C and $c \in C$. Then the residual code $Res(C, c)$ of C with respect to c is the code generated by the restriction of G to the columns where c has a zero entry.

Theorem

Suppose C is a binary $[n, k, d]$ code and suppose $c \in C$ has weight ω, where $d > \omega/2$. Then $Res(C, c)$ is an $[n - \omega, k - 1, d']$ code with $d' \geq d - \omega + \lceil \omega/2 \rceil$.

Preliminaries

Definition

Let G be a generator matrix of a linear binary $[n, k, d]$ code C and $c \in C$. Then the residual code $Res(C, c)$ of C with respect to c is the code generated by the restriction of G to the columns where c has a zero entry.

Theorem

Suppose C is a binary $[n, k, d]$ code and suppose $c \in C$ has weight ω, where $d > \omega/2$. Then $Res(C, c)$ is an $[n - \omega, k - 1, d']$ code with $d' \geq d - \omega + \lceil \omega/2 \rceil$.

Preliminaries

Theorem

Suppose C is a binary $[n, k, d]$ code with dual distance $d^{\perp}$, $c \in C$, and the dimension of $Res(C, c)$ is $k - 1$. Then the dual distance of $Res(C, c)$ is also $d^{\perp}$.

We use the program Q_EXTENSION to construct all inequivalent $[n, k, d]$ codes from their residual or shortening codes.

First approach:

Moving backwards through the residuals of a supposed $[n, k, d]^{d^\perp}$ code (where the superscript means that the code has dual distance $d^\perp$) we can extend as:

$[k_0, k_0, 1] \to [n_0, k_0, d_0]^{d^\perp} \to \ldots \to$

$\to [n - d, k - 1, \ \geq d/2]^{d^\perp} \to [n, k, d]^{d^\perp}$

(In fact, this does in most cases become a *tree* of extensions).

Computer programs

We use the program Q_EXTENSION to construct all inequivalent $[n, k, d]$ codes from their residual or shortening codes.

First approach:

Moving backwards through the residuals of a supposed $[n, k, d]^{d^\perp}$ code (where the superscript means that the code has dual distance $d^\perp$) we can extend as:

$$[k_0, k_0, 1] \rightarrow [n_0, k_0, d_0]^{d^\perp} \rightarrow \ldots \rightarrow$$

$$\rightarrow [n-d, k-1, \ \geq d/2]^{d^\perp} \rightarrow [n, k, d]^{d^\perp}$$

(In fact, this does in most cases become a *tree* of extensions).

Second approach:

We construct all $[n, k, d]$ codes by extending from their shortened codes.

I.e. from codes of the form $[n-i, k-i, d]$ or $[n-i-1, k-i, d]$.

If G is a generator matrix for an $[n-i, k-i, d]$ or an $[n-i-1, k-i, d]$ code we extend it in all possible ways to

$$\left(\begin{array}{c|c} * & \mathbf{I}_i \\ \hline \mathbf{G} & \mathbf{0} \end{array}\right) \quad \text{or} \quad \left(\begin{array}{c|c} * & \mathbf{1}\ \mathbf{I}_i \\ \hline \mathbf{G} & \mathbf{0} \end{array}\right).$$

Finding $N(9,5)$ and $N(10,5)$

From Brouwer's table we know that there may exist binary $[27,10,9]$ and $[28,10,10]$ codes with dual distance 5.

If we let C_{27} be a $[27,10,9]$ linear code with dual distance 5 we can consider a generator matrix of C_{27} in the form:

$$G_{27} = \left(\begin{array}{cc} 00000 & \\ \ldots & G_{22} \\ 00000 & \\ \hline 11000 & \\ 10100 & A \\ 10010 & \\ 10001 & \end{array} \right)$$

(where G_{22} generates a $[22,6,9]$ code).

Finding $N(9,5)$ and $N(10,5)$

From Brouwer's table we know that there may exist binary $[27,10,9]$ and $[28,10,10]$ codes with dual distance 5.

If we let C_{27} be a $[27,10,9]$ linear code with dual distance 5 we can consider a generator matrix of C_{27} in the form:

$$G_{27} = \left(\begin{array}{cc} 00000 & \\ \dots & G_{22} \\ 00000 & \\ \hline 11000 & \\ 10100 & A \\ 10010 & \\ 10001 & \end{array} \right)$$

(where G_{22} generates a $[22,6,9]$ code).

Finding $N(9,5)$ and $N(10,5)$

From Brouwer's table we know that there may exist binary $[27,10,9]$ and $[28,10,10]$ codes with dual distance 5.

If we let C_{27} be a $[27,10,9]$ linear code with dual distance 5 we can consider a generator matrix of C_{27} in the form:

$$G_{27} = \left(\begin{array}{cc} 00000 & \\ \dots & G_{22} \\ 00000 & \\ \hline 11000 & \\ 10100 & A \\ 10010 & \\ 10001 & \end{array} \right)$$

(where G_{22} generates a $[22,6,9]$ code).

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \rightarrow [23,6,10](29) \rightarrow [25,7,10](30522) \rightarrow [26,8,10](507533)$

$\rightarrow [27,9,10](30418) \rightarrow [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \to [23,6,10](29) \to [25,7,10](30522) \to [26,8,10](507533)$
$\to [27,9,10](30418) \to [28,10,10](10).$

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array}\right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \to [23,6,10](29) \to [25,7,10](30522) \to [26,8,10](507533)$
$\to [27,9,10](30418) \to [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \rightarrow [23,6,10](29) \rightarrow [25,7,10](30522) \rightarrow [26,8,10](507533)$
$\rightarrow [27,9,10](30418) \rightarrow [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \rightarrow [23,6,10](29) \rightarrow [25,7,10](30522) \rightarrow [26,8,10](507533)$
$\rightarrow [27,9,10](30418) \rightarrow [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \to [23,6,10](29) \to [25,7,10](30522) \to [26,8,10](507533)$

$\to [27,9,10](30418) \to [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \rightarrow [23,6,10](29) \rightarrow [25,7,10](30522) \rightarrow [26,8,10](507533)$

$\rightarrow [27,9,10](30418) \rightarrow [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Adding a parity check bit to G_{27} we obtain a generator matrix of a code C_{28} with parameters $[28,10,10]$. This generator matrix has the form:

$$G_{28} = \left(\begin{array}{ccc} 00000 & & \\ \dots & G_{23} & \\ 00000 & & \\ \hline 11000 & & b_7 \\ 10100 & A & b_8 \\ 10010 & & b_9 \\ 10001 & & b_{10} \end{array} \right)$$

(where G_{23} generates a [23,6,10] code).

By exhaustive search we find all inequivalent $[28,10,10]$ codes.

The extensions are:

$[6,6,1] \to [23,6,10](29) \to [25,7,10](30522) \to [26,8,10](507533)$

$\to [27,9,10](30418) \to [28,10,10](10)$.

Finding $N(9,5)$ and $N(10,5)$

Out of these ten, five turn out to have dual distance 5.

$N(10,5) = 28$ with 5 inequivalent codes.

By deleting each coordinate and analysing the results, we find that there are exactly 137 inequivalent $[27,10,9]$ codes with dual distance 5.

$N(9,5) = 27$ with 137 inequivalent codes.

Finding $N(9,5)$ and $N(10,5)$

Out of these ten, five turn out to have dual distance 5.

$N(10,5) = 28$ with 5 inequivalent codes.

By deleting each coordinate and analysing the results, we find that there are exactly 137 inequivalent $[27,10,9]$ codes with dual distance 5.

$N(9,5) = 27$ with 137 inequivalent codes.

N(12,6)

Extensions:

$[5,5,1] \to [15,5,\geq 6](91) \to [27,6,12](178) \to [28,7,12](129) \to [29,8,12](73) \to [30,9,12](9) \to [31,10,12](2) \to [32,11,12](2).$

The $[32,11,12]$ codes turn out to have dual distance 6, which is optimal in the sence that no shorter code, or with different dimension, could achieve this.

Moreover, the $[31,10,12]$ codes turn out to have dual distance 5, which is also optimal.

$N(12,5) = 31$ and $N(12,6) = 32$.

Puncturing (and optimality) give us $N(11,6) = 31$, $N(10,6) = 30$ and $N(9,6) = 29$. And also $N(11,5) = 30$.

N(12,6)

Extensions:

$[5,5,1] \to [15,5,\geq 6](91) \to [27,6,12](178) \to [28,7,12](129) \to [29,8,12](73) \to [30,9,12](9) \to [31,10,12](2) \to [32,11,12](2)$.

The $[32,11,12]$ codes turn out to have dual distance 6, which is optimal in the sence that no shorter code, or with different dimension, could achieve this.

Moreover, the $[31,10,12]$ codes turn out to have dual distance 5, which is also optimal.

$N(12,5) = 31$ and $N(12,6) = 32$.

Puncturing (and optimality) give us $N(11,6) = 31$, $N(10,6) = 30$ and $N(9,6) = 29$. And also $N(11,5) = 30$.

N(12,6)

Extensions:

$[5,5,1] \rightarrow [15,5,\geq 6](91) \rightarrow [27,6,12](178) \rightarrow [28,7,12](129) \rightarrow [29,8,12](73) \rightarrow [30,9,12](9) \rightarrow [31,10,12](2) \rightarrow [32,11,12](2)$.

The $[32,11,12]$ codes turn out to have dual distance 6, which is optimal in the sence that no shorter code, or with different dimension, could achieve this.

Moreover, the $[31,10,12]$ codes turn out to have dual distance 5, which is also optimal.

$N(12,5) = 31$ and $N(12,6) = 32$.

Puncturing (and optimality) give us $N(11,6) = 31$, $N(10,6) = 30$ and $N(9,6) = 29$. And also $N(11,5) = 30$.

N(12,6)

Extensions:

$[5,5,1] \to [15,5,\geq 6](91) \to [27,6,12](178) \to [28,7,12](129) \to [29,8,12](73) \to [30,9,12](9) \to [31,10,12](2) \to [32,11,12](2)$.

The $[32,11,12]$ codes turn out to have dual distance 6, which is optimal in the sence that no shorter code, or with different dimension, could achieve this.

Moreover, the $[31,10,12]$ codes turn out to have dual distance 5, which is also optimal.

$N(12,5) = 31$ and $N(12,6) = 32$.

Puncturing (and optimality) give us $N(11,6) = 31$, $N(10,6) = 30$ and $N(9,6) = 29$. And also $N(11,5) = 30$.

N(12,6)

Extensions:

$[5,5,1] \to [15,5,\geq 6](91) \to [27,6,12](178) \to [28,7,12](129) \to [29,8,12](73) \to [30,9,12](9) \to [31,10,12](2) \to [32,11,12](2)$.

The $[32,11,12]$ codes turn out to have dual distance 6, which is optimal in the sence that no shorter code, or with different dimension, could achieve this.

Moreover, the $[31,10,12]$ codes turn out to have dual distance 5, which is also optimal.

$N(12,5) = 31$ and $N(12,6) = 32$.

Puncturing (and optimality) give us $N(11,6) = 31$, $N(10,6) = 30$ and $N(9,6) = 29$. And also $N(11,5) = 30$.

Table of the $N(d, d^{\perp})$ function

$d/d^{\perp}$	3	4	5	6	7	8	9	10	11	12
3	6 (1)	-	-	-	-	-	-	-	-	-
4	7 (1)	8 (1)	-	-	-	-	-	-	-	-
5	11 (1)	13 (1)	16 (1)	-	-	-	-	-	-	-
6	12 (1)	14 (1)	17 (1)	18 (1)	-	-	-	-	-	-
7	14 (1)	15 (1)	20 (1)	21 (1)	22* (1)	-	-	-	-	-
8	15 (1)	16 (1)	21 (1)	22* (1)	23* (1)	24* (1)	-	-	-	-
9	20 (3)	22 (1)	27 (137)	29 ($\geq$ 2)	32-37	33-41	38-42	-	-	-
10	21 (2)	24 (2)	28 (5)	30 ($\geq$ 2)	33-41	34-42	39-43	40-44	-	-
11	23 (1)	26 (1)	30 (2)	31 (2)	36-42	37-43	41-44	43-45	46* (1)	-
12	24 (1)	28 (7)	31 (2)	32 (2)	37-43	38-44	42-45	44-46	47* (1)	48* (1)

Thank you for your attention!