An $\mathcal{O}(n)$ Algorithm for Coefficient Grouping

Fukang Liu

University of Hyogo, Hyogo, Japan
liufukangs@gmail.com

Abstract. In this note, we study a specific optimization problem arising in the recently proposed coefficient grouping technique, which is used for the degree evaluation. Specifically, we show that there exists an efficient algorithm running in time $\mathcal{O}(n)$ to solve a basic optimization problem relevant to upper bound the algebraic degree. We expect that some results in this note can inspire more studies on other optimization problems in the coefficient grouping technique.

Keywords: coefficient grouping, optimization problem

1 Notation

The following notations will be used throughout this paper.

1. $a\%b$ represents a mod b.
2. $a|b$ denotes that a divides b.
3. $[a, b]$ is a set of integers i satisfying $a \leq i \leq b$.
4. $H(a)$ is the hamming weight of $a \in [0, 2^n - 1]$.
5. The function $\mathcal{M}_n(x)$ ($x \geq 0$) is defined as follows:

$$\mathcal{M}_n(x) = \begin{cases} 2^n - 1 \text{ if } 2^n - 1|x, x \geq 2^n - 1, \\ x\%(2^n - 1) \text{ otherwise.} \end{cases}$$

By the definition of $\mathcal{M}_n(x)$, we have $\mathcal{M}_n(x_1 + x_2) = \mathcal{M}_n(\mathcal{M}_n(x_1) + \mathcal{M}_n(x_2))$, $\mathcal{M}_n(2^i) = 2^{i\%n}$ and $\mathcal{M}_n(2^i x) = \mathcal{M}_n(2^{i\%n}\mathcal{M}_n(x))$ for $i \geq 0$.

2 Motivation

We have recently developed a technique called coefficient grouping to upper bound the algebraic degree for ciphers defined over $\mathbb{F}_{2^n}$. The main idea of that technique is to convert the degree evaluation into some optimization problems. Among them, one basic optimization problem can be described as follows:

$$\begin{aligned} &\text{maximize} \ \ H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)), \\ &\text{subject to} \ \ \gamma_i \in \mathbb{N}, 0 \leq \gamma_i \leq N_i \text{ for } i \in [0, n-1], \end{aligned}$$

where $(N_{n-1}, N_{n-2}, \ldots, N_0)$ is a known vector of nonnegative integers. Note that throughout this paper, we always consider integers and hence we omit $\gamma_i \in \mathbb{N}$ later.

In [1], this problem is first encoded as an MILP problem and then solved with an off-the-shelf solver Gurobi. Using a general-purpose blackbox solver is indeed very convenient but we may lose some insight into this special problem.

Regarding why we do not put this note in [1], we just cannot find a good place. First, we feel it not suitable to place this short note at the Appendix of [1] as few people may read it and then neglect its importance. Placing it at the main content of [1] also looks inappropriate because it may destroy the simplicity and structure of [1]. The most important reason is that we can only find an efficient algorithm for one specific optimization problem, while there are several different optimization problems in [1] and they all can be handled by solvers.

One purpose of this note is thus to share our ideas of one specific optimization problem and we expect that they can inspire more studies. The technique in this note is of independent interest.

3 An Efficient Algorithm for the Optimization Problem

Our aim is to solve the following optimization problem when given a vector of nonnegative integers $(N_{n-1}, N_{n-2}, \ldots, N_0)$:

$$\begin{aligned} &\text{maximize} \ \ H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)), \\ &\text{subject to} \ \ 0 \le \gamma_i \le N_i \text{ for } i \in [0, n-1]. \end{aligned}$$

Or equivalently, we want to find an element e with the maximal hamming weight from the following set

$$\mathcal{S} = \{e | e = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i), 0 \le \gamma_i \le N_i \text{ for } i \in [0, n-1]\}.$$

In this note, we show an efficient algorithm to solve the above optimization problem in time $\mathcal{O}(n)$, as shown in Algorithm 1. In the following, we mainly focus on how to prove its correctness.

Lemma 1 *If there exists an index i such that $N_i \ge 2^n - 1$, the solution to the above problem is directly n. Moreover, if $N_i \ge 1$ for all $i \in [0, n-1]$, the solution to the above problem is also n.*

Proof. For both cases, we can trivially find an assignment to $(\gamma_{n-1}, \gamma_{n-2}, \ldots, \gamma_0)$ such that

$$2^n - 1 = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i).$$

Algorithm 1 Compute the solution to the optimization problem

```
procedure DEGREE(N_{n-1}, N_{n-2}, ..., N_0)
    finish = 0
    while finish = 0 do
        finish = 1
        nonzero = 1
        for i in range (n) do
            if N_i ≥ 2^n − 1 then
                return n
            else if N_i ≥ 3 then
                finish = 0
            else if N_i = 0 then
                nonzero = 0
        if nonzero = 1 then
            return n
        if finish = 0 then
            for i in range (n) do
                if N_i%2 = 1 then
                    N_{(i+1)%n} = N_{(i+1)%n} + (N_i − 1)/2
                    N_i = 1
                else if N_i > 0 and N_i%2 = 0 then
                    N_{(i+1)%n} = N_{(i+1)%n} + (N_i − 2)/2
                    N_i = 2
    d = 0
    for i in range (n) do
        if N_i > 0 then
            d++
    return d
```

Hence, we find an assignment to make $H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i)) = n$. As n is the upper bound for the solution, the solution to this optimization problem is n. $\square$

Theorem 1 ***(Equivalence.)*** *Let $(N'_{n-1}, N'_{n-2}, \ldots, N'_0)$ and $(N_{n-1}, N_{n-2}, \ldots, N_0)$ be two vectors of nonnegative integers such that $N'_i = N_i$ for $i \in \mathcal{I} = \{0, 1, \ldots, n-1\} \setminus \{j, (j+1)\%n\}$ and $N_j > 0$. Moreover, when $N_j\%2 = 1$,*

$$\begin{cases} N'_j = 1, \\ N'_{(j+1)\%n} = \dfrac{N_j - 1}{2} + N_{(j+1)\%n}. \end{cases} \tag{1}$$

When $N_j\%2 = 0$,

$$\begin{cases} N'_j = 2, \\ N'_{(j+1)\%n} = \dfrac{N_j - 2}{2} + N_{(j+1)\%n}. \end{cases} \tag{2}$$

Then, for

$$\mathcal{S}_1 = \{e | e = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i), 0 \le \gamma_i \le N_i \text{ for } i \in [0, n-1]\},$$
$$\mathcal{S}_2 = \{e | e = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i), 0 \le \gamma_i \le N'_i \text{ for } i \in [0, n-1]\},$$

we have $\mathcal{S}_1 = \mathcal{S}_2$.

Proof. For the given index j where $N_j > 0$ and $(N_j, N_{(j+1)\%n}, N'_j, N'_{(j+1)\%n})$ satisfying either Equation 1 or Equation 2, we first prove that $\mathcal{S}_3 = \mathcal{S}_4$, where

$$\mathcal{S}_3 = \{e | e = 2^j a + 2^{j+1} b, 0 \le a \le N_j, 0 \le b \le N_{(j+1)\%n}\},$$
$$\mathcal{S}_4 = \{e | e = 2^j a + 2^{j+1} b, 0 \le a \le N'_j, 0 \le b \le N'_{(j+1)\%n}\}.$$

When $(N_j, N_{j+1}, N'_j, N'_{j+1})$ satisfies Equation 1, we have

$$\begin{aligned} \mathcal{S}_3 &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N_j + 2^{j+1} \cdot N_{(j+1)\%n}\}, \\ \mathcal{S}_4 &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N'_j + 2^{j+1} \cdot N'_{(j+1)\%n}\} \\ &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot 1 + 2^{j+1} \cdot (\frac{N_j - 1}{2} + N_{(j+1)\%n})\} \\ &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N_j + 2^{j+1} \cdot N_{(j+1)\%n}\} \\ &= \mathcal{S}_3. \end{aligned}$$

When $(N_j, N_{j+1}, N'_j, N'_{j+1})$ satisfies Equation 2, we have

$$\begin{aligned} \mathcal{S}_3 &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N_j + 2^{j+1} \cdot N_{(j+1)\%n}\}, \\ \mathcal{S}_4 &= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N'_j + 2^{j+1} \cdot N'_{(j+1)\%n}\} \end{aligned}$$

$$= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot 2 + 2^{j+1} \cdot (\frac{N_j - 2}{2} + N_{(j+1)\%n})\}$$
$$= \{e | 2^j \cdot 0 + 2^{j+1} \cdot 0 \le e \le 2^j \cdot N_j + 2^{j+1} \cdot N_{(j+1)\%n}\}$$
$$= \mathcal{S}_3.$$

Hence, $\mathcal{S}_3 = \mathcal{S}_4$ is proved. As $\mathcal{S}_1$ and $\mathcal{S}_2$ can also be represented as follows:

$$\mathcal{S}_1 = \{e | e = \mathcal{M}_n(e_0 + \sum_{i \in \mathcal{I}} 2^i \gamma_i), 0 \le \gamma_i \le N_i \text{ for } i \in \mathcal{I}, e_0 \in \mathcal{S}_3\},$$
$$\mathcal{S}_2 = \{e | e = \mathcal{M}_n(e_1 + \sum_{i \in \mathcal{I}} 2^i \gamma_i), 0 \le \gamma_i \le N_i \text{ for } i \in \mathcal{I}, e_1 \in \mathcal{S}_4\},$$

we have $\mathcal{S}_1 = \mathcal{S}_2$. □

3.1 Explaining Our Algorithm

The correctness of Algorithm 1 highly relies on the consecutive applications of Theorem 1. Specifically, in the loop from Line 16 $-$ Line 22, we always find an index j such that $N_j > 0$ and then convert $(N_{n-1}, N_{n-2}, \ldots, N_0)$ in the following way.

When N_j is an odd number, we do the following conversion:

$$(N_{n-1}, \ldots, N_{(j+1)\%n}, N_j, \ldots, N_0) \leftarrow \ldots, N_{(j+1)\%n} + \frac{N_j - 1}{2}, 1, \ldots, N_0).$$

When N_j is an even number, we do the following conversion:

$$(N_{n-1}, \ldots, N_{(j+1)\%n}, N_j, \ldots, N_0) \leftarrow (N_{n-1}, \ldots, N_{(j+1)\%n} + \frac{N_j - 2}{2}, 2, \ldots, N_0)$$

Let us denote the output vector after the loop by $(N'_{n-1}, N'_{n-2}, \ldots, N'_0)$. Based on Theorem 1, the original optimization problem is reduced to an equivalent optimization problem:

$$\begin{aligned} &\text{maximize} \;\; H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)), \\ &\text{subject to} \;\; 0 \le \gamma_i \le N'_i \text{ for } i \in [0, n-1]. \end{aligned}$$

This is because $\mathcal{S}_1 = \mathcal{S}_2$ where

$$\mathcal{S}_1 = \{e | e = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i), 0 \le \gamma_i \le N_i \text{ for } i \in [0, n-1]\},$$
$$\mathcal{S}_2 = \{e | e = \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i), 0 \le \gamma_i \le N'_i \text{ for } i \in [0, n-1]\}.$$

Moreover, the output vector $(N'_{n-1}, N'_{n-2}, \ldots, N'_0)$ must be of the following 4 possible forms:

Form 1: $\exists i \in [0, n-1]$, $N'_i \geq 2^n - 1$.
Form 2: $\forall i \in [0, n-1]$, $N'_i > 0$.
Form 3: $\forall i \in [0, n-1]$, $N'_i \in [0, 2]$ and $\exists j \in [0, n-1]$, $N'_j = 0$.
Form 4: $\forall i \in [1, n-1]$, $N'_i \in [0, 2]$, $2 < N'_0 < 2^n - 1$ and $\exists j \in [0, n-1]$, $N'_j = 0$.

For the first two forms, according to Lemma 1, the solution to the equivalent optimization problem is n and hence the solution to the original optimization problem is also n. This corresponds to Line 7 $-$ Line 8 and Line 13 $-$ Line 14 of Algorithm 1.

For Form 3, we will terminate the `While` loop and computes the number of nonzero elements in $(N'_{n-1}, N'_{n-2}, \ldots, N'_0)$ denoted by d.

For Form 4, we will again move to the loop from Line 16 $-$ Line 22. Since the input vector $(N_{n-1}, N_{n-2}, \ldots, N_0)$ now satisfies

$$\forall i \in [1, n-1], N_i \in [0, 2], 2 < N_0 < 2^n - 1 \text{ and } \exists j \in [0, n-1], N_j = 0,$$

the output vector after this loop, which is still denoted by $(N'_{n-1}, N'_{n-2}, \ldots, N'_0)$, must be of Form 2 or 3.

Hence, we are left to prove that the solution to the following optimization problem is d when there are d nonzero elements in the vector $(N_{n-1}, N_{n-2}, \ldots, N_0)$ where $\forall i \in [0, n-1], N_i \in [0, 2]$:

$$\begin{aligned} &\text{maximize } \ H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)), \\ &\text{subject to } \ 0 \leq \gamma_i \leq N_i \text{ for } i \in [0, n-1]. \end{aligned}$$

If this is proved, the correctness of Line 23 $-$ Line 27 is proved and hence the correctness of Algorithm 1 is proved. Moreover, according to the above analysis, it runs in time $\mathcal{O}(n)$. In the following, we focus on the proof.

Lemma 2 *For any $a, b \in [0, 2^n - 1]$, we have $H(\mathcal{M}_n(a + b)) \leq H(a) + H(b)$.*

Proof. Let $(a_{n-1}, a_{n-2}, \ldots, a) \in \mathbb{F}_2^n$ and $(b_{n-1}, b_{n-2}, \ldots, b_0) \in \mathbb{F}_2^n$ be the binary representations of a and b, respectively. Let $\mathcal{I}_0 = \{i_{0,1}, i_{0,2}, \ldots, i_{0,p_0}\}$ and $\mathcal{I}_1 = \{i_{1,1}, j_{1,2}, \ldots, i_{1,p_1}\}$ be the sets of indices such that $a_i = 1$ and $b_j = 1$ for $i \in \mathcal{I}_0$ and $j \in \mathcal{I}_1$. In other words, $H(a) = p_0$ and $H(b) = p_1$. Let

$$\mathcal{I}_2 = \mathcal{I}_0 \cap \mathcal{I}_1 = \{i_{2,1}, i_{2,2}, \ldots, i_{2,p_2}\}.$$

Then, we have

$$p_2 \leq min\{p_0, p_1\}.$$

In this way, we have

$$\mathcal{M}_n(a + b) = \mathcal{M}_n(\sum_{i \in \mathcal{I}_0 \setminus \mathcal{I}_2} 2^i + \sum_{i \in \mathcal{I}_1 \setminus \mathcal{I}_2} 2^i + 2 \sum_{i \in \mathcal{I}_2} 2^i) = \mathcal{M}_n(\alpha_3 + \alpha_4),$$

$$\alpha_3 = \sum_{i\in\mathcal{I}_0\setminus\mathcal{I}_2} 2^i + \sum_{i\in\mathcal{I}_1\setminus\mathcal{I}_2} 2^i,$$
$$\alpha_4 = \sum_{i\in\mathcal{I}_2} 2^{(i+1)\%n}.$$

Hence, we have

$$H(\alpha_3) = p_0 + p_1 - 2p_2,$$
$$H(\alpha_4) = p_2 \le min\{p_0, p_1\}.$$

Repeating the same analysis, i.e. for $k \ge 1$, let

$$\mathcal{I}_{3k} = (\mathcal{I}_{3(k-1)} \cup \mathcal{I}_{3(k-1)+1}) \setminus \mathcal{I}_{3(k-1)+2} = \{i_{3k,1}, i_{3k,2}, \ldots, i_{3k,p_{3k}}\},$$
$$\mathcal{I}_{3k+1} = \{j | j = (i+1)\%n, i \in \mathcal{I}_{3(k-1)+2}\} = \{i_{3k+1,1}, i_{3k+1,2}, \ldots, i_{3k+1,p_{3k+1}}\},$$
$$\mathcal{I}_{3k+2} = \mathcal{I}_{3k} \cap \mathcal{I}_{3k+1} = \{i_{3k+2,1}, i_{3k+2,2}, \ldots, i_{3k+2,p_{3k+2}}\}.$$

Then, we have

$$\begin{aligned}
\mathcal{M}_n(a+b) &= \mathcal{M}_n(\alpha_{3k} + \alpha_{3k+1}) \\
&= \mathcal{M}_n(\sum_{i\in\mathcal{I}_{3k}\setminus\mathcal{I}_{3k+2}} 2^i + \sum_{i\in\mathcal{I}_{3k+1}\setminus\mathcal{I}_{3k+2}} 2^i + 2\sum_{i\in\mathcal{I}_{3k+2}} 2^i) \\
&= \mathcal{M}_n(\alpha_{3(k+1)} + \alpha_{3(k+1)+1}), \\
\alpha_{3(k+1)} &= \sum_{i\in\mathcal{I}_{3k}\setminus\mathcal{I}_{3k+2}} 2^i + \sum_{i\in\mathcal{I}_{3k+1}\setminus\mathcal{I}_{3k+2}} 2^i, \\
\alpha_{3(k+1)+1} &= \sum_{i\in\mathcal{I}_{3k+2}} 2^{(i+1)\%n}.
\end{aligned}$$

Moreover,

$$\begin{aligned}
p_{3(k+1)} &= p_{3k} + p_{3k+1} - 2p_{3k+2}, \\
p_{3(k+1)+1} &= p_{3k+2}, \\
p_{3(k+1)+2} &\le min\{p_{3k} + p_{3k+1} - 2p_{3k+2}, p_{3k+2}\} \le p_{3k+2}, \\
p_{3(k+1)} + p_{3(k+1)+1} &\le p_{3k} + p_{3k+1} \le \ldots \le p_0 + p_1.
\end{aligned}$$

Therefore, $p_{3(k+1)+2} \le p_{3k+2} \le \ldots \le p_2 \le min\{p_0, p_1\}$ must hold. Moreover, it is impossible to have a sequence $p_{3(s+\ell)+2} = \cdots = p_{3(s+1)+2} = p_{3s+2} > 0$ for $s \ge 0$ and $\ell \ge p_0 + p_1$. If there is, we have

$$\begin{aligned}
p_{3(s+\ell)+2} &= p_{3(s+\ell-1)+2} \le min\{p_{3(s+\ell-1)} + p_{3(s+\ell-1)+1} - 2p_{3(s+\ell-1)+2}, p_{3(s+\ell-1)+2}\} \\
&\Rightarrow p_{3(s+\ell-1)} + p_{3(s+\ell-1)+1} \ge 3p_{3(s+\ell-1)+2} = 3p_{3(s+\ell-2)+2} \\
&\Rightarrow p_{3(s+\ell-2)} + p_{3(s+\ell-2)+1} - p_{3(s+\ell-2)+2} \ge 3p_{3(s+\ell-2)+2} \\
&\Rightarrow p_{3(s+\ell-2)} + p_{3(s+\ell-2)+1} \ge 4p_{3(s+\ell-2)+2} = 4p_{3(s+\ell-3)+2} \\
&\Rightarrow p_{3(s+\ell-3)} + p_{3(s+\ell-3)+1} - p_{3(s+\ell-3)+2} \ge 4p_{3(s+\ell-3)+2}
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow p_{3(s+\ell-3)} + p_{3(s+\ell-3)+1} \geq 5p_{3(s+\ell-3)+2} = 5p_{3(s+\ell-4)+2} \\
&\Rightarrow \cdots \\
&\Rightarrow p_{3(s+1)} + p_{3(s+1)+1} \geq (\ell+1)p_{3s+2} \geq \ell+1 \geq p_0 + p_1 + 1
\end{aligned}$$

However, we also have $p_{3(s+1)} + p_{3(s+1)+1} \leq p_0 + p_1$, which causes a contradiction. Therefore, p_{3k+2} cannot always remain the same value and it must decrease at some k. Hence, there must exist $\hat{k}$ such that $p_{3\hat{k}+2} = 0$, i.e. $\mathcal{I}_{3\hat{k}} \cap \mathcal{I}_{3\hat{k}+1} = \emptyset$. In particular, in this case, we have

$$\mathcal{M}_n(a+b) = \mathcal{M}_n(\alpha_3 + \alpha_4) = \cdots = \mathcal{M}_n(\alpha_{3\hat{k}} + \alpha_{3\hat{k}+1}) = \alpha_{3\hat{k}} + \alpha_{3\hat{k}+1}.$$

As $H(\alpha_{3\hat{k}}) = p_{3\hat{k}}$, $H(\alpha_{3\hat{k}+1}) = p_{3\hat{k}+1}$, $\mathcal{I}_{3\hat{k}} \cap \mathcal{I}_{3\hat{k}+1} = \emptyset$ and $p_{3\hat{k}} + p_{3\hat{k}+1} \leq p_0 + p_1$, we have $H(\mathcal{M}_n(a+b)) = p_{3\hat{k}} + p_{3\hat{k}+1} \leq p_0 + p_1 = H(a) + H(b)$. $\square$

Theorem 2 *For any $m_1, m_2, \ldots, m_t \in [0, 2^n - 1]$, we have*

$$H(\mathcal{M}_n(m_1 + m_2 + \cdots + m_t)) \leq H(m_1) + H(m_2) + \cdots + H(m_t).$$

Proof. According to Lemma 2, we have

$$\begin{aligned}
& H(\mathcal{M}_n(m_1 + m_2 + \cdots + m_t)) \\
= \ & H(\mathcal{M}_n(m_1 + \mathcal{M}_n(\sum_{i=2}^{t} m_i))) \\
\leq \ & H(m_1) + H(\mathcal{M}_n(\sum_{i=2}^{t} m_i)) \\
\leq \ & H(m_1) + H(m_2) + H(\mathcal{M}_n(\sum_{i=3}^{t} m_i)) \\
\cdots & \\
\leq \ & H(m_1) + H(m_2) + \cdots + H(m_t).
\end{aligned}$$

$\square$

Theorem 3 *Let $(N_{n-1}, N_{n-2}, \ldots, N_0)$ be such a vector that $\forall i \in [0, n-1], N_i \in [0, 2]$ and it has in total d nonzero elements. Then, the solution to the following optimization problem*

$$\begin{aligned}
&\text{maximize} \ \ H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)), \\
&\text{subject to} \ \ 0 \leq \gamma_i \leq N_i \text{ for } i \in [0, n-1]
\end{aligned}$$

is d.

Proof. Let $\mathcal{J} = \{j_1, j_2, \ldots, j_d\}$ be the set of indices such that $N_j > 0$ for $j \in \mathcal{J}$ and $N_j = 0$ for $j \notin \mathcal{J}$.

Since

$$N_i \in \{0,1,2\} \text{ for } i \in [0, n-1],$$

for each $(\gamma_{n-1}, \gamma_{n-2}, \ldots, \gamma_0)$ satisfying $\gamma_i \leq N_i$ for $i \in [0, n-1]$, we have $H(\mathcal{M}_n(2^j \gamma_j)) \leq 1$ for $j \in \mathcal{J}$ and $H(\mathcal{M}_n(2^j \gamma_j)) = 0$ for $j \notin \mathcal{J}$. According to Theorem 2, we immediately obtain

$$\begin{aligned}
&H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)) \\
&= H(\mathcal{M}_n(\sum_{i=0}^{n-1} \mathcal{M}_n(2^i \gamma_i))) \leq \sum_{i=0}^{n-1} H(\mathcal{M}_n(2^i \gamma_i)) = \sum_{j \in \mathcal{J}} H(\mathcal{M}_n(2^j \gamma_j)) \leq d.
\end{aligned}$$

In other words, the upper bound for the solution to the optimization problem is d. By making $\gamma_j = 1$ for $j \in \mathcal{J}$, we have

$$H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)) = d.$$

In other words, we find an assignment to make the solution to the optimization problem be d. Hence, the solution to the optimization problem is d. $\square$

References

1. F. Liu, R. Anand, L. Wang, W. Meier, and T. Isobe. Coefficient Grouping: Breaking Chaghri and More. 2022. `https://eprint.iacr.org/2022/???`