NAIST-IS-DT9761027

Doctor's Thesis

Optimality and Ruling-Out Conditions and Their Evaluation Methods for Soft-Decision Iterative Decoding Algorithms for Binary Linear Block Codes

Yuansheng Tang

February 8, 2000

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Yuansheng Tang

Thesis committee: Toru Fujiwara, Professor
Masaki Koyama, Professor
Hiroyuki Seki, Professor
Yuichi Kaji, Associate Professor

Optimality and Ruling-Out Conditions and Their Evaluation Methods for Soft-Decision Iterative Decoding Algorithms for Binary Linear Block Codes*

Yuansheng Tang

Abstract

For the decoding of binary block codes, many iterative soft-decision decoding algorithms have been derived. Most of these algorithms are devised to find the best (or the most likely) codeword in a series of candidates which are usually generated by means of simple decoders, such as an algebraic decoder, in successive iterative steps by use of information on the reliability measures of the received sequence. To terminate the iterative process in time, a termination condition is used to restrict the number of iterative steps or tested at the end of each iterative step. For maximum-likelihood decoding, the termination condition is usually chosen as an optimality condition. The decoder used internally may fail to generate new candidate codewords (decoding failure or repetition) at some iterative steps. Hence, the execution of these iterative steps results unnecessary decoding operations and prolongs the decoding delay. Usually, at the end of each iterative step, some more conditions are tested to control the iterative process without degrading the error performance of the algorithm. If one of the conditions is satisfied, then, in a few successive iterative steps which are prearranged by the algorithm, the internal decoder can not generate any candidate codeword which may be better than the best candidate codeword obtained so far, and thus these successive iterative steps can be skipped. Such a condition is called a ruling-out condition.

This thesis mainly deals with the design and the evaluation of optimality conditions and ruling-out conditions for iterative soft-decision decoding algorithms.

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9761027, February 8, 2000.

At first, a sufficient condition, denoted $\text{Cond}_{\text{opt},h}$, for the optimality of a decoded codeword based on h reference words are derived for any iterative soft-decision decoding algorithms for binary block codes which are based on the generation of a series of codeword candidates.

Next, for some concrete iterative soft-decision decoding algorithms, such as Chase-type decoding algorithms, some generalizations of Chase algorithm, the iterative minimum weight trellis search (MWTS) decoding algorithm, GMD-like decoding and multistage GMD-like decoding, some more effective optimality conditions and some ruling-out conditions are proposed.

Then, a class of nonlinear integer programming problems (IPPs) which can be used to evaluate the proposed testing conditions are shown. For some particular cases, the IPPs are solved completely. Especially, the IPP related to the optimality condition $\text{Cond}_{\text{opt},h}$, denoted $\mathcal{P}(Q_h^*)$, is investigated. The IPP $\mathcal{P}(Q_h^*)$ has been solved for $h \leq 3$ in literature. For $h \leq 5$, it is shown that the IPP $\mathcal{P}(Q_h^*)$ can be split uniformly into a few simpler sub-IPPs, the number of variables of each sub-IPP is only half of that of the original IPP. By use of this splitting, an effective algorithm for solving the IPP $\mathcal{P}(Q_4^*)$ is given. The number of operations of additions and comparisons of real numbers, which is the dominant factor of the computational complexity of the algorithm, is shown to be of order N^2, where N is the length of the code.

An improvement of the optimality condition $\text{Cond}_{\text{opt},h}$ is proposed further by use of partial information of the distance profile of the code. Some methods for evaluating the improved optimality condition for some particular cases are shown.

The first five search centers of the multistage Chase-type decoding algorithm that consists of a series of Chase-type decodings for which the next search center is chosen as the best one among the words that have not been visited are also determined. The number of operations of additions and comparisons of real numbers for the determination of the search centers is shown to be of order τ, where τ is the number of errors that can be corrected in an iterative step of a Chase-type decoding.

Keywords:

Binary block code, iterative soft-decision decoding, maximum-likelihood decoding, optimality condition, ruling-out condition, integer programming problem.

Acknowledgments

I have been fortunate to have received support and assistance from many individuals. I would especially like to thank Professor Toru Fujiwara and Emeritus Professor Tadao Kasami for their invaluable support, discussions, and encouragement throughout the work.

I am grateful to Professors Masaki Koyama and Hiroyuki Seki for their invaluable support on the work. I am also grateful to Associate Professor Yuichi Kaji for his invaluable support and suggestions.

I am grateful to Professor Toyoo Takata of Iwate Prefectural University for his invaluable support on the work.

I would like to thank Dr. Hajime Watanabe and Dr. Takuya Koumoto for their helpful support and discussions.

Lastly, I am grateful to all the members of Fujiwara Laboratory of Nara Institute of Science and Technology.

List of Publications

1 Papers Related to the Work

1.1 Journal Papers

[1] Tadao Kasami, Yuansheng Tang, Takuya Koumoto, and Toru Fujiwara, "Sufficient Conditions for Ruling-Out Useless Iterative Steps in a Class of Iterative Decoding Algorithm," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science, vol.E82-A, no.10, pp.2061–2073, October 1999.

[2] Yuansheng Tang, Toru Fujiwara and Tadao Kasami, "The Sectionalized Trellises with the Smallest Computational Complexity of the Generalized Version of Viterbi Algorithm of Linear Block Codes," submitted for publication to IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science.

1.2 International Conference Paper

[3] Yuansheng Tang, Tadao Kasami, and Toru Fujiwara, "An Optimality Testing Algorithm for a Decoded Codeword of Binary Block Codes and Its Computational Complexity," Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: Proceedings of the 13th International Symposium, AAECC-13, Honolulu, Hawaii, USA, Lecture Notes in Computer Science, vol.1719, pp.201–210, Spring Verlag, November 1999.

1.3 Other Conference Papers

[4] Yuansheng Tang and Tadao Kasami, "On a Testing Condition on the Optimality of a Decoded Codeword of Binary Block Codes," Proceedings of the 20th Symposium on Information Theory and Its Applications, pp.321–324, Matsuyama, Japan, December 1997.

[5] Yuansheng Tang, Tadao Kasami, and Toru Fujiwara, "The Sectionalized Trellises with Fewest Edges of Linear Block Codes," Technical Report of IEICE, IT97-84, March 1998.

[6] Yuansheng Tang, Tadao Kasami, and Toru Fujiwara, "An Optimality Testing Algorithm for a Decoded Codeword of Binary Block Codes and Its Computational Complexity," Proceedings of the 21st Symposium on Information Theory and Its Applications, pp.499–502, Gifu, Japan, December 1998.

1.4 Technical Reports of NAIST

[7] Yuansheng Tang and Tadao Kasami, "One Sufficient Termination Condition for Some Iterative Soft-Decision Algorithms Decoding Binary Linear Block Codes," Technical Report of NAIST, NAIST-IS-TR97012, June 1997.

[8] Yuansheng Tang, Tadao Kasami and Toru Fujiwara, "An Optimality Testing Algorithm for a Decoded Codeword of Binary Block Codes," Technical Report of NAIST, NAIST-IS-TR98012, October 1998.

2 Other Journal Papers

[9] Yuansheng Tang, "On the Blichfildt Lemma," Journal of Qingdao University (PRC), vol.4, no.3–4, pp.96–98, February 1991 (in Chinese).

[10] Yuansheng Tang, "On the Stability of the Linear Systems with Variational Coefficients," Journal of Qingdao University(PRC), vol.6, no.1, pp.19–24, February 1993 (in Chinese).

[11] Pincong Shao and Yuansheng Tang, "The Analytic Solutions of a Class of Functional Integral Equations," Journal of Binzhou Teachers College(PRC), vol.9, no.2, pp.4–9, May 1993 (in Chinese).

[12] Yuansheng Tang, "On a Kind of Measurement Property of the Continued Fractions," Journal of Shandong Polytechnic University (PRC), suppl., pp.18–21, November 1993 (in Chinese).

[13] Yuansheng Tang, "The Local Properties of the Monotone Functions and Convex Functions," Journal of Qingdao University (PRC), vol.6, no.4, pp.1–9, November 1993 (in Chinese).

[14] Pincong Shao and Yuansheng Tang, "On the Regularity Properties of Additive Functions," Acta Mathematica Sinica, New series, vol.10, Special Issue, pp.180–188, January 1994.

[15] Yuansheng Tang, "On the Even Solutions of Single Kurtosis of the Feigenbaum Function Equations," Journal of Qingdao University (PRC), vol.7, no.1, pp.29–35, February 1994 (in Chinese).

[16] Yuansheng Tang, "On the Mean-Value of the Enumerational Function of the Multiplicative Partitions," Journal of Mathematics (PRC), vol.14, no.2, pp.211–216, May 1994 (in Chinese).

[17] Yuansheng Tang, "A Generalization of Poincare-Bendixson Theorem," Journal of Qingdao University (PRC), vol.7, no.2, pp.1–8, May 1994 (in Chinese).

[18] Yuansheng Tang, "A Property of R-Space in Arithmetic Functions," Journal of Qingdao University (PRC), vol.7, no.3, pp.46–50, August 1994 (in Chinese).

[19] Yuansheng Tang, "The Kátai-Property of Real Number Pairs $\langle a, b\rangle$ and an Application to Multiplicative Functions," Journal of Qingdao University (PRC), vol.8, no.4, pp.36–48, November 1995 (in Chinese).

[20] Eduard Wirsing, Yuansheng Tang and Pintsung Shao (Pincong Shao), "On a Conjecture of Kátai for Additive Functions," Journal of Number Theory, vol.56, no.2, pp.391–395, February 1996.

[21] Yuansheng Tang and Pincong Shao, "The Proof of a Conjecture of Kátai," Acta Mathematica Sinica, vol.39, no.2, pp.190–195, March 1996 (in Chinese).

[22] Yuansheng Tang, "A Reverse Problem on Arithmetic Functions," Journal of Number Theory, vol.58, no.1, pp.130–138, May 1996.

[23] Yuansheng Tang, "On the C^∞-Solutions of Feigenbaum Function Equations," Acta Mathematica Sinica, vol.40, no.1, pp.253–258, January 1997 (in Chinese).

[24] Yuansheng Tang, "A Property of R-Space in Arithmetic Functions, II," Journal of Qingdao University (PRC), vol.10, no.1, pp.23–30, February 1997 (in Chinese).

[25] Yuansheng Tang, "On the Convex-Solutions of the Feigenbaum Function Equations," Journal of Applied Mathematics (PRC), vol.15, no.1, pp.34–41, February 1997 (in Chinese).

[26] Yuansheng Tang, "On the Regularity Properties of Multiplicative Functions, I," Mathematica Japonica, vol.48, no.3, pp.409–417, October 1998.

[27] Yuansheng Tang, "On the Regularity Properties of Multiplicative Functions, II," Mathematica Japonica, vol.48, no.3, pp.419–426, October 1998.

Contents

Chapter 1

Introduction

The decoding of binary block codes is a very important problem both in coding theory and practical data transmission. So far, many **iterative soft-decision decoding algorithms** have been developed [1-12], [17-27], etc. Most of these algorithms are devised to find the best (or the most likely) codeword in a series of candidates which are usually generated with simple decoders, such as an algebraic decoder, in successive iterative steps by use of information on the reliability measures of the received sequence. The generalized minimum distance (GMD) decoding algorithm introduced by Forney [2] uses a series of at most $(d_{\min}+1)/2$ algebraic errors-and-erasures decoding trials to generate the codeword candidates, where $d_{\min}$ is the minimum Hamming distance of the code. For the Chase algorithm [1], a bounded distance decoding is implemented around some search centers that are words obtained by adding the hard-decision tuple to the error patterns whose nonzero components are confined to those bit positions with less reliability. In recent years, these algorithms have been improved and applied by many theorists in [3-12], [17-27], etc.

To terminate the iterative process in time, a **termination condition** is used to restrict the number of iterative steps or tested at the end of each iterative step. If the iterative soft-decision decoding algorithm is **maximum-likelihood decoding** (MLD), which decodes a received sequence to the optimal (or the most likely) codeword, the termination condition is usually chosen as an **optimality condition** which may be renewed in the iterative process. At the end of an iterative step, if an optimality condition is satisfied, then the best codeword candidate obtained so far is the optimal codeword and the iterative process is terminated. An effective optimality condition can provide fast termination to the iterative process of an iterative soft-decision maximum-likelihood decoding algorithm.

In general, for an iterative soft-decision decoding algorithm, the decoder used internally may fail to generate new codeword candidates (decoding failure or repetition) at some iterative steps. Hence, the execution of these iterative steps prolongs the decoding delay. Therefore, it is desired to reduce the iterative steps as many as possible and the decoding complexity without degrading the error performance of the algorithm. Usually, at the end of each iterative step, some more suitably devised conditions are tested to control the iterative process. **Ruling-out conditions** and **early termination conditions** are two classes of such kind of testing conditions which have been used widely.

If a ruling-out condition is satisfied, then, in a few successive iterative steps which are prearranged by the algorithm, the internal decoder can not generate any codeword candidate which may be better than the best codeword candidate obtained so far, and thus these successive iterative steps can be skipped. An early termination condition can be viewed as the strongest version of a ruling-out condition. If an early termination condition is satisfied, then there is no improvement on error performance by any further iteration, and thus the iterative process is terminated.

It is worthwhile and possible to devise effective testing conditions for many well-known iterative soft-decision decoding algorithms. This thesis mainly deals with the design and the computational complexity analysis of testing conditions for iterative soft-decision decoding algorithms.

In Chapter 2, an optimality condition, denoted $\text{Cond}_{\text{opt},h}$, based on h reference words are derived for any iterative soft-decision decoding algorithms for binary block codes which are based on the generation of a series of codeword candidates. The simplified acceptance criterion introduced by Taipale and Pursley [25] for GMD decoding is the simplest particular case of this optimality condition with $h = 1$. In general, the larger the number h of the reference words, the more powerful the optimality condition. By use of the search region which has been searched or will be searched in the next a few steps by the iterative decoding algorithm, a general optimality condition and a general ruling-out condition are proposed. We address a Chase-type decoding algorithm as an iterative soft-decision decoding algorithm which consists of a series of bounded distance decodings implemented around some words obtained by adding a given word, called search center, to the error patterns in a test set that is pre-determined or chosen in the iterative process. Some effective testing conditions are also devised for a Chase-type decoding algorithm if the test set is the pre-determined set of the error patterns whose nonzero components are confined to a few least reliable bit positions. The testing conditions used in [11] and [12] are particular cases of those proposed in this chapter. For some generalizations of Chase algorithm, the iterative minimum weight trellis search (MWTS) decoding algorithm, GMD-like decoding and multistage GMD-like decoding, some effective optimality conditions are also proposed.

In Chapter 3, the testing conditions proposed in Chapter 2 are formulated uniformly into the optimal value of a class of nonlinear **integer programming problems (IPPs)**, which can also be applied in many other iterative soft-decision decoding algorithms for binary block codes. For some particular cases, the IPPs are solved completely.

For the optimality condition $\mathrm{Cond}_{\mathbf{opt},h}$, the related IPP, denoted $\mathcal{P}(Q_h^*)$, is considered in Chapter 4. For $h \leq 3$, the IPP $\mathcal{P}(Q_h^*)$ has been solved in [17]-[19], and applied effectively in [21]. Since the object function of $\mathcal{P}(Q_h^*)$ is nonlinear and has 2^h variables, there is no known method which solves it effectively for large h. For $1 \leq h \leq 5$, we split the IPP $\mathcal{P}(Q_h^*)$ into a few simple sub-IPPs, the number of variables of each sub-IPP is only half of that of the original IPP. Then some effective algorithms for solving the IPP $\mathcal{P}(Q_h^*)$ for $1 \leq h \leq 4$ are shown. For the case $h = 4$, the sub-IPPs are split further into some simpler sub-IPPs which can be solved simply by iteration. The number of operations of additions and comparisons of real numbers, which is the dominant factor of the computational complexity of the algorithm, is shown to be of order N for $h \leq 3$, and of order N^2 for $h = 4$, where N is the length of the code. It turns out that $\mathrm{Cond}_{\mathbf{opt},4}$ can also reduce significantly the average decoding complexity and provide fast termination of the decoding iteration without degrading the error performance as $\mathrm{Cond}_{\mathbf{opt},h}$ with $h \leq 3$ does. An improvement of the optimality condition $\mathrm{Cond}_{\mathbf{opt},h}$ is proposed further by use of partial information of the distance profile of the code. Some evaluation methods for some particular cases of the improved optimality condition are shown.

A multistage Chase-type decoding algorithm, consisting of several stages of Chase-type decodings, is investigated in Chapter 5. The first stage of Chase-type decoding is the original Chase decoding. At the end of each stage of Chase-type decoding, the search center of the next stage of Chase-type decoding is chosen as the best one among the words that have not been searched by the Chase-type decodings implemented already [13]. The first three search centers have already been determined. In this chapter, two algorithms for determining the fourth and fifth search centers are presented, respectively. For these two algorithms, the numbers of operations of additions and comparisons of real numbers, which are also the dominant factor of computational complexity of the algorithms, are shown to be of order τ, where τ is the number of errors that can be corrected in an iterative step of a Chase-type decoding.

Chapter 2

Testing Conditions for Soft-Decision Iterative Decoding Algorithms

2.1 Definitions and Notations

Let $V \triangleq \mathrm{GF}(2)$. For convenience, the two elements 0 and 1 in V will be viewed as integers when they are multiplied by integers or real numbers. For a positive integer N, let V^N denote the set of all binary sequences $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$ of length N. Suppose a binary block code $C \subseteq V^N$ with minimum distance $d_{\min}$ is used for error control over the additive white Gaussian noise (AWGN) channel with binary-phase-shift-keyed (BPSK) signaling. Assume that each codeword is transmitted with the same probability. Before the transmission of a codeword $\boldsymbol{c} = (c_1, c_2, \ldots, c_N) \in C$, $\boldsymbol{c}$ is mapped into the bipolar sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ with $x_j \triangleq 2c_j - 1 \in \{1, -1\}$. At the output of a matched filter in the receiver, the **received sequence** related to the codeword $\boldsymbol{c}$ is $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ with $r_i = x_i + w_i$, where for $1 \leq i \leq N$, w_i's are statistically independent Gaussian random variables with zero mean. Let $\boldsymbol{z} = (z_1, z_2, \ldots, z_N)$ be the binary **hard-decision sequence** which is a binary sequence obtained from $\boldsymbol{r}$ by

$$z_j \triangleq \begin{cases} 1, & \text{if } r_j > 0, \\ 0, & \text{if } r_j \leq 0, \end{cases} \quad j = 1, 2, \ldots, N.$$

For $1 \leq j \leq N$, the reliability of hard-decision symbol z_j can be measured by the magnitude $|r_j|$ since it is proportional to the log-likelihood ratio associated with z_j.

For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, the correlation between $\boldsymbol{u}$ and the received sequence $\boldsymbol{r}$ is given by

$$M(\boldsymbol{u}) = \sum_{i=1}^{N}(2u_i - 1)r_i. \tag{2.1}$$

For a codeword $\boldsymbol{c}$ in C, the larger the correlation $M(\boldsymbol{c})$ is, the larger the probability of $\boldsymbol{c}$ to be the transmitted codeword.

For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, define

$$\mathcal{D}_1(\boldsymbol{u}) \triangleq \{i : u_i \neq z_i,\ 1 \leq i \leq N\}, \tag{2.2}$$

$$\mathcal{D}_0(\boldsymbol{u}) \triangleq \{1, 2, \ldots, N\} \setminus \mathcal{D}_1(\boldsymbol{u}), \tag{2.3}$$

$$L(\boldsymbol{u}) \triangleq \sum_{i \in \mathcal{D}_1(\boldsymbol{u})} |r_i|. \tag{2.4}$$

It follows from (2.1), (2.2) and (2.4) that $M(\boldsymbol{u})$ can be expressed in terms of $M(\boldsymbol{z})$ and $L(\boldsymbol{u})$ as

$$\begin{aligned} M(\boldsymbol{u}) &= \sum_{i=1}^{N}(2u_i - 1)r_i \\ &= \sum_{i=1}^{N}(2z_i - 1)r_i - \sum_{i=1}^{N}(2z_i - 2u_i)r_i \\ &= M(\boldsymbol{z}) - 2L(\boldsymbol{u}). \end{aligned}$$

$L(\boldsymbol{u})$ is called the **correlation discrepancy** of $\boldsymbol{u}$ with respect to hard-decision sequence $\boldsymbol{z}$. For any subset T of V^N, let $\underline{L}[T] \triangleq \min_{\boldsymbol{u} \in T} L(\boldsymbol{u})$, and write $\underline{L}[\phi] \triangleq +\infty$ for convenience.

The **maximum likelihood decoding** (MLD) may be stated in terms of the correlation discrepancy as: The decoder decodes the received sequence $\boldsymbol{r}$ into the **optimal codeword** $\boldsymbol{c}_{\text{opt}} \in C$ with $L(\boldsymbol{c}_{\text{opt}}) = \underline{L}[C]$. If $\boldsymbol{z}$ is a codeword, then $\boldsymbol{z}$ is the optimal codeword since $L(\boldsymbol{u}) \geq 0 = L(\boldsymbol{z})$ for any $\boldsymbol{u} \in V^N$.

Let IDA denote a **soft-decision iterative decoding algorithm** for C. If the binary hard-decision sequence $\boldsymbol{z}$ is in C, then $\boldsymbol{z}$ is output as the decoded codeword and the decoding process terminates. Otherwise, IDA starts from the first iterative step. At the j-th iterative step, IDA performs the following two sub-procedures (G) and (T):

(G) Generation of a codeword candidate: A region $R(j)$ is searched by an internal decoder to find the best codeword $\boldsymbol{c}(j)$ with

$$L(\boldsymbol{c}(j)) = \underline{L}[R(j) \cap C]. \tag{2.5}$$

The codeword $\boldsymbol{c}(j)$ is called the **codeword candidate** generated at the j-th iterative step. If $R(j) \cap C = \phi$, no codeword candidate is generated at the j-th iterative step. In this case, $\boldsymbol{c}(j)$ is undefined, and is denoted $*$. For convenience, we define $L(*) \triangleq +\infty$. Let

$$R_{\mathrm{p}}(j) \triangleq \bigcup_{i=1}^{j} R(i). \tag{2.6}$$

For convenience, define

$$R_{\mathrm{p}}(\infty) \triangleq \bigcup_{i=1}^{\infty} R(i). \tag{2.7}$$

A codeword candidate $\boldsymbol{c}$ is said to be **better** than another codeword candidate $\boldsymbol{c}'$ if $L(\boldsymbol{c}) < L(\boldsymbol{c}')$. Among a specified set of candidate codewords, a codeword candidate $\boldsymbol{c}$ is said to be the **best** if $L(\boldsymbol{c})$ achieves the minimum.

Let $\boldsymbol{c}_{\mathrm{best}}(j)$ be the best codeword in $R_{\mathrm{p}}(j) \cap C$ with

$$L(\boldsymbol{c}_{\mathrm{best}}(j)) = \underline{L}[R_{\mathrm{p}}(j) \cap C]. \tag{2.8}$$

We denote $\boldsymbol{c}_{\mathrm{best}}(j) = *$ if $R_{\mathrm{p}}(j) \cap C = \phi$.

(T) Test of termination condition: If a new candidate codeword is generated, then a termination condition, abbreviated to $\mathrm{Cond}_{\mathrm{T}}$, is tested. If $\mathrm{Cond}_{\mathrm{T}}$ is satisfied, then $\boldsymbol{c}_{\mathrm{best}}(j)$ is output as the decoded codeword and the decoding process terminates (if '$*$' is output, IDA fails in decoding). If a new codeword candidate is not generated or $\mathrm{Cond}_{\mathrm{T}}$ is not satisfied, IDA proceeds to the $(j+1)$-th iterative step. □

In the end of this section, we introduce some notations which are used in the remainder of the thesis.

For $1 \leq i \leq j \leq N$, define $[i,j] \triangleq \{i, i+1, \ldots, j\}$. For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$, $\boldsymbol{v} = (v_1, v_2, \ldots, v_N) \in V^N$ and a nonempty set $I \subseteq [1, N]$, define

$$d_{\mathrm{H},I}(\boldsymbol{u}, \boldsymbol{v}) \triangleq |\{i \in I : u_i \neq v_i\}|. \tag{2.9}$$

For $I = \{i_1, i_2, \ldots, i_m\}$ with $1 \leq i_1 < i_2 < \cdots < i_m \leq N$,

$$p_I(\boldsymbol{u}) \triangleq (u_{i_1}, u_{i_2}, \ldots, u_{i_m}). \tag{2.10}$$

We abbreviate $d_{\mathrm{H},[1,N]}(\boldsymbol{u}, \boldsymbol{v})$, $d_{\mathrm{H},[i,j]}(\boldsymbol{u}, \boldsymbol{v})$ and $p_{[i,j]}(\boldsymbol{u})$ to $d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v})$, $d_{\mathrm{H},i,j}(\boldsymbol{u}, \boldsymbol{v})$ and $p_{i,j}(\boldsymbol{u})$, respectively.

For simplicity, hereafter throughout the thesis, we assume that the bit positions $1, 2, \ldots, N$ are ordered according to the reliability order given as follows:

$$|r_1| \leq |r_2| \leq \cdots \leq |r_N|. \tag{2.11}$$

For convenience, we define $r_\infty \triangleq +\infty$. For any subset $X \subseteq \{1, 2, \ldots, N, \infty\}$ and integer j, let $X^{(j)}$ denote the set of j smallest integers in $X \backslash \{\infty\}$ if $1 \leq j \leq |X \backslash \{\infty\}|$, the empty set ϕ if $j \leq 0$, the set $X \cup \{\infty\}$ if $j > |X \backslash \{\infty\}|$, respectively.

For $\boldsymbol{u} \in V^N$ and a positive integer d, define

$$O_d(\boldsymbol{u}) \triangleq \{\boldsymbol{v} \in V^N : d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{u}) \leq d\}, \tag{2.12}$$

$$\bar{O}_d(\boldsymbol{u}) \triangleq \{\boldsymbol{v} \in V^N : d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{u}) \geq d\}. \tag{2.13}$$

2.2 General Optimality Condition

If the IDA is an MLD, the termination condition $\mathrm{Cond}_{\mathrm{T}}$ is usually chosen as an **optimality condition**, denoted $\mathrm{Cond}_{\mathrm{opt}}$. If $\mathrm{Cond}_{\mathrm{opt}}$ is satisfied at some iterative step, then the best candidate codeword obtained so far is the optimal codeword. An effective $\mathrm{Cond}_{\mathrm{opt}}$ can provide fast termination to the iterative process. Below we first give a general $\mathrm{Cond}_{\mathrm{opt}}$, which can be incorporated in any IDA that is based on the generation of a series of codeword candidates. Then a $\mathrm{Cond}_{\mathrm{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(j)$ with use of $R_{\mathrm{p}}(j)$ is shown.

Let $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\ldots$, $\boldsymbol{u}_h$, called **reference words**, be h words in V^N, and let $d_1, d_2, \ldots, d_h$, called reference distances, be h positive integers. Now we give a

Cond_{opt} based on these reference words and reference distances. Let

$$
\begin{aligned}
V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h) &\triangleq V^N \setminus \left(\bigcup_{j=1}^{h} O_{d_j-1}(\boldsymbol{u}_j)\right) \\
&= \bigcap_{j=1}^{h} \bar{O}_{d_j}(\boldsymbol{u}_j) \\
&= \{\boldsymbol{v} \in V^N : d_{\text{H}}(\boldsymbol{u}_j,\boldsymbol{v}) \geq d_j \text{ for } 1 \leq j \leq h\}. \quad (2.14)
\end{aligned}
$$

We have the following simple lemma.

Lemma 2.1: If a codeword $\boldsymbol{c}_{\text{best}}$ in $\bigcup_{j=1}^{h} O_{d_j-1}(\boldsymbol{u}_j)$ satisfies

$$
L(\boldsymbol{c}_{\text{best}}) \leq \underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)], \quad (2.15)
$$

then the optimal codeword $\boldsymbol{c}_{\text{opt}}$ must belong to the set $\bigcup_{j=1}^{h} O_{d_j-1}(\boldsymbol{u}_j)$. Furthermore, we have $\boldsymbol{c}_{\text{best}} = \boldsymbol{c}_{\text{opt}}$ if

$$
L(\boldsymbol{c}_{\text{best}}) = \underline{L}\left[C \cap \left(\bigcup_{j=1}^{h} O_{d_j-1}(\boldsymbol{u}_j)\right)\right]. \quad (2.16)
$$

Lemma 2.1 not only defines a region which contains the optimal codeword $\boldsymbol{c}_{\text{opt}}$ but also provides a Cond_{opt} for the candidates which have been generated. It can be incorporated in any IDA which is based on the generation of a series of codeword candidates. However, the effectiveness of Lemma 2.1 is significantly influenced by the selection of the reference words and the reference distances. In general, we can select the reference words from the previously generated candidate codewords, which are trusted to possess low correlation discrepancies, and take "the covering distance around $\boldsymbol{u}_j$" assured by the IDA as the reference distance related to the reference word $\boldsymbol{u}_j$.

The optimality condition (2.15) will also be referred as $\text{Cond}_{\text{opt},h}$. In general, the larger the h, the stronger the optimality condition $\text{Cond}_{\text{opt},h}$. However, as the number of the reference words grows significantly, the computational complexity for evaluating the lower bound $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)]$ grows significantly. Thus, only the most recent candidate and those possessing larger covering distances or smaller correlation discrepancies are chosen as reference words. The evaluation of the lower bound $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)]$ will be considered in Chapter 4.

Next we consider to give a Cond_{opt} for $\boldsymbol{c}_{\text{best}}(j)$ with use of $R_{\text{p}}(j)$.

Lemma 2.2: Let U be a subset of $R_p(j) \cap C$. If $c_{best}(j) \neq *$, then

$$L(c_{best}(j)) \leq \underline{L}[(V^N \setminus R_p(j)) \cap V(U)] \tag{2.17}$$

is a Cond_{opt} for $c_{best}(j)$, where

$$V(U) \triangleq \{u \in V^N : d_H(u, c) \geq d_{min} \text{ for all } c \in U\}. \tag{2.18}$$

Proof: For any $X \subseteq V^N$, it follows from $U \subseteq R_p(j) \cap C$ and $C \setminus U \subseteq V(U)$ that

$$\begin{aligned}(X \setminus R_p(j)) \cap C &= (X \setminus R_p(j)) \cap (C \setminus (R_p(j) \cap C)) \\ &\subseteq (X \setminus R_p(j)) \cap (C \setminus U) \\ &\subseteq (X \setminus R_p(j)) \cap V(U).\end{aligned} \tag{2.19}$$

If the condition (2.17) holds, then from (2.19) with $X = V^N$,

$$\begin{aligned}\underline{L}[C] &= \min\{\underline{L}[(V^N \setminus R_p(j)) \cap C], \underline{L}[R_p(j) \cap C]\} \\ &\geq \min\{\underline{L}[(V^N \setminus R_p(j)) \cap V(U)], L(c_{best}(j))\} \\ &= L(c_{best}(j)),\end{aligned}$$

i.e. $c_{best}(j)$ is the optimal codeword. □

In Sections 2.4 to 2.7, the Cond_{opt}s (2.15) and (2.17) will be considered further for some concrete IDAs.

2.3 General Ruling-Out Condition

Sometimes, the IDA may fail to generate a new candidate codeword (decoding failure or repetition) at an iterative step. Hence, the execution of this kind of iterative steps results unnecessary decoding operations and prolongs the decoding delay. Therefore, it is desired to reduce such iterative steps as many as possible. At the end of the j-th iterative step, for some $j' > j$, a (j, j')**-ruling-out (useless iterative steps) condition**, denoted Cond_{RO}, is tested. If Cond_{RO} is satisfied, then between the $(j+1)$-th iterative step and the j'-th one, the internal decoder can not generate any codeword candidate which may be better than $c_{best}(j)$, and thus these iterative steps may be skipped. A $(j, +\infty)$-ruling-out condition is also called **early termination condition**, denoted Cond_{ET}.

The following lemma gives a Cond_{RO} by use of the set $R_p(j') \setminus R_p(j)$.

Lemma 2.3: For $j' > j$ and $U \subseteq R_p(j) \cap C$,

$$L(c_{best}(j)) \leq \underline{L}[(R_p(j') \setminus R_p(j)) \cap V(U)] \tag{2.20}$$

is a (j, j')-ruling-out condition.

Proof: If the condition (2.20) holds, then for any codeword $c \in R_p(j')$, from (2.19) with $X = R_p(j')$, we have

$$\begin{aligned} L(c) &\geq \underline{L}[R_p(j') \cap C] \\ &= \min\{\underline{L}[(R_p(j') \setminus R_p(j)) \cap C], \underline{L}[R_p(j) \cap C]\} \\ &\geq \min\{\underline{L}[(R_p(j') \setminus R_p(j)) \cap V(U)], L(c_{best}(j))\} \\ &= L(c_{best}(j)). \end{aligned}$$

Hence the condition (2.20) is a (j, j')-ruling-out condition. □

In Section 2.4, the Cond_{RO} (2.20) will be considered further for Chase-Type Decoding Algorithm.

2.4 Testing Conditions for Chase-Type Decoding Algorithm

For a word $v \in V^N$ and positive integers t_0 and τ which satisfy $0 < t_0 \leq \lfloor (d_{min} - 1)/2 \rfloor$ and $1 \leq \tau \leq N - t_0$, a **Chase-type decoding algorithm**, denoted Chase(v, t_0, τ), is defined as an IDA which performs **bounded distance-t_0 decoding** around the word $v + e$ for every e whose nonzero components are confined to the first τ bit positions. e is called a test error pattern. Chase$(z, \lfloor (d_{min} - 1)/2 \rfloor, \lfloor d_{min}/2 \rfloor)$ is the original Chase decoding algorithm [1].

We consider to give some useful testing conditions for Chase(v, t_0, τ). For simplicity, we assume the test error patterns are generated in **binary order** and let $e(j)$ denote the j-th test error pattern with $1 \leq j \leq 2^\tau$. That is, $e(j) = (e(j)_1, e(j)_2, \ldots, e(j)_\tau, 0, \ldots, 0)$ is the N-tuple in V^N which satisfies

$$\sum_{i=1}^{\tau} e(j)_i 2^{i-1} = j - 1. \tag{2.21}$$

In the j-th iterative step, the **search region** is

$$R(j) = \{\boldsymbol{u} \in V^N : d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}(j)) \leq t_0\}, \tag{2.22}$$

where $\boldsymbol{v}(j) \triangleq \boldsymbol{v} + \boldsymbol{e}(j)$ is called the search center at the j-th iterative step.

We will use (2.17), (2.20) and the following lemma to give some concrete testing conditions for Chase$(\boldsymbol{v}, t_0, \tau)$.

Lemma 2.4: (1) For $1 \leq l \leq \tau + 1$, we have that

$$R_{\mathrm{p}}(2^{l-1}) = \{\boldsymbol{u} \in V^N : d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) \leq t_0\}. \tag{2.23}$$

(2) For $2^{l-1} < j \leq 2^l$ with $1 \leq l \leq \tau$, we have that

$$R(j) \setminus R_{\mathrm{p}}(2^{l-1}) = \{\boldsymbol{u} \in V^N : p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(j)), d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v}) = t_0\}. \tag{2.24}$$

(3) For $2^{l-1} \leq j' < j \leq 2^l$ with $1 \leq l \leq \tau$, we have that

$$(R(j) \setminus R_{\mathrm{p}}(2^{l-1})) \cap (R(j') \setminus R_{\mathrm{p}}(2^{l-1})) = \phi. \tag{2.25}$$

Proof: (1) From (2.21), we see that

$$\{\boldsymbol{e}(j) : 1 \leq j \leq 2^{l-1}\} = \{\boldsymbol{e} \in V^N : p_{l,N}(\boldsymbol{e}) = \boldsymbol{0}\}. \tag{2.26}$$

Proof of $\subseteq$. For $\boldsymbol{u} \in R(j)$ with $1 \leq j \leq 2^{l-1}$, it follows from (2.22) and (2.26) that

$$d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) = d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v} + \boldsymbol{e}(j)) \leq d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}(j)) \leq t_0,$$

and thus $\boldsymbol{u}$ belongs to the right-hand side of (2.23).
Proof of $\supseteq$. For any $\boldsymbol{u} \in V^N$ with $d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) \leq t_0$, from (2.26), we see that there is an integer j with $1 \leq j \leq 2^{l-1}$ such that

$$p_{1,l-1}(\boldsymbol{e}(j)) = p_{1,l-1}(\boldsymbol{u} + \boldsymbol{v}) \text{ and } p_{l,N}(\boldsymbol{e}(j)) = \boldsymbol{0},$$

and thus

$$d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v} + \boldsymbol{e}(j)) = d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) \leq t_0.$$

This means that $\boldsymbol{u} \in R(j)$, and thus $\boldsymbol{u} \in R_{\mathrm{p}}(2^{l-1})$ by $R(j) \subseteq R_{\mathrm{p}}(2^{l-1})$.
(2) Since $2^{l-1} < j \leq 2^l$, the test pattern $\boldsymbol{e}(j)$ satisfies

$$\boldsymbol{e}(j)_l = 1 \text{ and } p_{l+1,N}(\boldsymbol{e}(j)) = \boldsymbol{0}. \tag{2.27}$$

Proof of $\supseteq$. For any $\boldsymbol{u} \in V^N$ with

$$p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(j)) \text{ and } d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v}) = t_0,$$

from (2.27), we see that

$$d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}(j)) = d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v} + \boldsymbol{e}(j)) = d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v}) = t_0,$$
$$d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) = d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v}) + d_{\mathrm{H},l,l}(\boldsymbol{v}(j), \boldsymbol{v}) = t_0 + 1.$$

Hence $\boldsymbol{u}$ belongs to $R(j) \setminus R_{\mathrm{p}}(2^{l-1})$ by (2.22) and (2.23).
Proof of $\subseteq$. For any $\boldsymbol{u} \in R(j) \setminus R_{\mathrm{p}}(2^{l-1})$, from (2.22), (2.27) and (2.23), we see that

$$t_0 \geq d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{v}(j)) = d_{\mathrm{H},1,l}(\boldsymbol{u}, \boldsymbol{v}(j)) + d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v})$$
$$\geq d_{\mathrm{H},1,l}(\boldsymbol{u}, \boldsymbol{v}(j)) + d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) - 1 \geq d_{\mathrm{H},1,l}(\boldsymbol{u}, \boldsymbol{v}(j)) + t_0.$$

Hence, $d_{\mathrm{H},1,l}(\boldsymbol{u}, \boldsymbol{v}(j)) = 0$ and $d_{\mathrm{H},l+1,N}(\boldsymbol{u}, \boldsymbol{v}) = t_0$, and then $\boldsymbol{u}$ belongs to the right-hand side of (2.24).
(3) For $2^{l-1} \leq j < j' \leq 2^l$, from (2.21) and (2.26), we have

$$\sum_{i=1}^{l} e(j)_i 2^{i-1} = j - 1 \neq j' - 1 = \sum_{i=1}^{l} e(j')_i 2^{i-1},$$

and thus $p_{1,l}(\boldsymbol{e}(j)) \neq p_{1,l}(\boldsymbol{e}(j'))$, it is equivalent to

$$p_{1,l}(\boldsymbol{v}(j)) \neq p_{1,l}(\boldsymbol{v}(j')). \tag{2.28}$$

Then (2.25) follows from (2.24) and (2.28). □

2.4.1 Optimality Conditions for Chase-Type Decoding Algorithm

For $1 \leq l \leq \tau + 1$, from (2.23),

$$V^N \setminus R_{\mathrm{p}}(2^{l-1}) = \{\boldsymbol{u} \in V^N : d_{\mathrm{H},l,N}(\boldsymbol{u}, \boldsymbol{v}) \geq t_0 + 1\}. \tag{2.29}$$

If U is a subset of $R_p(2^{l-1}) \cap C$, from (2.17) and (2.29),

$$L(c_{\text{best}}(2^{l-1})) \le \underline{L}[\{\boldsymbol{u} \in V^N : d_{\text{H},l,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0 + 1\} \cap V(U)] \tag{2.30}$$

is a Cond_{opt} for $c_{\text{best}}(2^{l-1})$.

Now we consider to give some Cond_{opt}s for $c_{\text{best}}(j)$ for $2^{l-1} < j < 2^l$ with $1 \le l \le \tau$. Since we have

$$R_p(j) = R_p(2^{l-1}) \cup \left(\bigcup_{2^{l-1} < i \le j} R(i) \right),$$

from (2.22) and (2.23), we see that

$$\begin{aligned} V^N \setminus R_p(j) = \{\boldsymbol{u} \in V^N &: d_{\text{H},l,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0 + 1, \\ & d_{\text{H}}(\boldsymbol{u},\boldsymbol{v}(i)) \ge t_0 + 1 \text{ for } 2^{l-1} < i \le j\}. \end{aligned} \tag{2.31}$$

If U is a subset of $R_p(j) \cap C$, then from (2.17) and (2.31),

$$\begin{aligned} L(c_{\text{best}}(j)) \le \underline{L}[\{\boldsymbol{u} \in V^N &: d_{\text{H},l,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0 + 1, \\ & d_{\text{H}}(\boldsymbol{u},\boldsymbol{v}(i)) \ge t_0 + 1 \text{ for } 2^{l-1} < i \le j\} \cap V(U)] \end{aligned} \tag{2.32}$$

is a Cond_{opt} for $c_{\text{best}}(j)$. If U contains some codeword candidates $\boldsymbol{c}(i)$ such that $2^{l-1} < i \le j$, then we can ignore these i's in the right side of (2.32) without effecting its value. Also, from (2.23) to (2.25), we see that

$$\begin{aligned} V^N \setminus R_p(j) &= (V^N \setminus R_p(2^l)) \cup (R_p(2^l) \setminus R_p(j)) \\ &= \{\boldsymbol{u} \in V^N : d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0 + 1\} \cup \\ &\quad \bigcup_{j < i \le 2^l} \{\boldsymbol{u} \in V^N : p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(i)), d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\}, \end{aligned} \tag{2.33}$$

and then we get another version of (2.32):

$$\begin{aligned} L(c_{\text{best}}(j)) \le \min\{&\underline{L}[\{\boldsymbol{u} \in V^N : d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0 + 1\} \cap V(U)], \\ &\min_{j < i \le 2^l} \{\underline{L}[\{\boldsymbol{u} \in V^N : p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(i)), d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]\}\}. \end{aligned} \tag{2.34}$$

We notice that the Cond_{opt} given by (2.32) (or the Cond_{opt} given by (2.34)) is suitable only for the cases that $(j - 2^{l-1})$ (or $(2^l - j)$) is a small integer, otherwise, the computational complexity of the evaluation of the right-hand side of (2.32) (or (2.34)) may be very large.

If $2^{l-1} < j \leq 2^{l-1} + 2^{l'-1}$ holds for some integer l' with $1 \leq l' < l$, then (2.34) can be written in a new form which may be evaluated more easily. Since an integer i satisfies $2^{l-1} + 2^{l'-1} < i \leq 2^l$ if and only if $\boldsymbol{e}(i)$ satisfies

$$p_{l',l-1}(\boldsymbol{e}(i)) \neq \boldsymbol{0}, \ \ e(i)_l = 1 \text{ and } p_{l+1,N}(\boldsymbol{e}(i)) = \boldsymbol{0},$$

from (2.33), we have that

$$\begin{aligned} V^N \setminus R_p(j) = &\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \geq t_0 + 1\} \cup \\ &\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l',l-1}(\boldsymbol{u},\boldsymbol{v}) \geq 1, u_l \neq v_l, d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cup \\ \bigcup_{j<i\leq 2^{l-1}+2^{l'-1}} &\{\boldsymbol{u} \in V^N : p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(i)), d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\}, \end{aligned}$$

and thus, for any subset U of $R_p(j) \cap C$,

$$\begin{aligned} L(\boldsymbol{c}_{\mathrm{best}}(j)) \leq \min\{&\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \geq t_0 + 1\} \cap V(U)], \\ &\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l',l-1}(\boldsymbol{u},\boldsymbol{v}) \geq 1, u_l \neq v_l, d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)], \\ &\min_{j<i\leq 2^{l-1}+2^{l'-1}} \{\underline{L}[\{\boldsymbol{u} \in V^N : p_{1,l}(\boldsymbol{u}) = p_{1,l}(\boldsymbol{v}(i)), \\ &\qquad d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]\}\} \end{aligned} \tag{2.35}$$

is a $\mathrm{Cond_{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(j)$ with $2^{l-1} < j \leq 2^{l-1} + 2^{l'-1}$ and $1 \leq l' < l$.

The $\mathrm{Cond_{opt}}$ given by (2.35) is suitable for the cases that the positive integer $2^{l-1} + 2^{l'-1} - j$ is very small. Especially, if $j = 2^{l-1} + 2^{l'-1}$, then (2.35) gives the following $\mathrm{Cond_{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(2^{l-1} + 2^{l'-1})$:

$$\begin{aligned} L(\boldsymbol{c}_{\mathrm{best}}(2^{l-1} + 2^{l'-1})) \leq \min\{&\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \geq t_0 + 1\} \cap V(U)], \\ &\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l',l-1}(\boldsymbol{u},\boldsymbol{v}) \geq 1, u_l \neq v_l, d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]\}. \end{aligned} \tag{2.36}$$

Clearly, for any $i \leq j$ and subset U of $R_p(j) \cap C$,

$$L(\boldsymbol{c}_{\mathrm{best}}(j)) \leq \underline{L}[(V^N \setminus R_p(i)) \cap V(U)] \tag{2.37}$$

is also a $\mathrm{Cond_{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(j)$.

According to (2.37), we can get some useful $\mathrm{Cond_{opt}}$s further. For example, if $2^{l-1}+2^{l'-1} \le j < 2^{l-1}+2^{l'}$ for some l' with $1 \le l' < l$, then for any subset U of $R_{\mathrm{p}}(j) \cap C$,

$$\begin{aligned} &L(\boldsymbol{c}_{\mathrm{best}}(j)) \le \underline{L}[(V^N \setminus R_{\mathrm{p}}(2^{l-1}+2^{l'-1})) \cap V(U)] \\ &= \min\{\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0+1\} \cap V(U)], \\ &\quad \underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l',l-1}(\boldsymbol{u},\boldsymbol{v}) \ge 1, u_l \ne v_l, d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]\} \end{aligned} \tag{2.38}$$

is a $\mathrm{Cond_{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(j)$.

For any j with $2^{l-1} \le j < 2^l$ and subset U of $R_{\mathrm{p}}(j) \cap C$,

$$\begin{aligned} L(\boldsymbol{c}_{\mathrm{best}}(j)) &\le \underline{L}[(V^N \setminus R_{\mathrm{p}}(2^{l-1})) \cap V(U)] \\ &= \underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l,N}(\boldsymbol{u},\boldsymbol{v}) \ge t_0+1\} \cap V(U)] \end{aligned} \tag{2.39}$$

is also a $\mathrm{Cond_{opt}}$ for $\boldsymbol{c}_{\mathrm{best}}(j)$.

2.4.2 Ruling-Out Conditions for Chase-Type Decoding Algorithm

For integers l_1, l_2 and l with $0 \le l_1 < l_2 < l \le \tau$, we see that an integer i satisfies $2^{l-1}+2^{l_1} < i \le 2^{l-1}+2^{l_2}$ if and only if $p_{l_1+1,l_2}(\boldsymbol{e}(i))$ is not equal to the all zero (l_2-l_1)-tuple $(0,0,\ldots,0)$ and

$$p_{l_2+1,N}(\boldsymbol{e}(i)) = p_{l_2+1,N}(\boldsymbol{e}(2^{l-1}+1)) = p_{l_2+1,N}(\boldsymbol{e}(2^{l-1}+2^{l_2})).$$

Hence, from (2.24) and (2.25), we see that

$$\begin{aligned} R_{\mathrm{p}}(2^{l-1}+2^{l_2}) \setminus R_{\mathrm{p}}(2^{l-1}+2^{l_1}) = \{&\boldsymbol{u} \in V^N : d_{\mathrm{H},l_1+1,l_2}(\boldsymbol{u},\boldsymbol{v}) \ge 1, \\ &p_{l_2+1,l}(\boldsymbol{u}) = p_{l_2+1,l}(\boldsymbol{v}(2^{l-1}+2^{l_2})), d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\}. \end{aligned} \tag{2.40}$$

If U is a subset of $R_{\mathrm{p}}(2^{l-1}+2^{l_1}) \cap C$, then from (2.20) and (2.40),

$$\begin{aligned} L(\boldsymbol{c}_{\mathrm{best}}(2^{l-1}+2^{l_1})) \le \underline{L}[\{&\boldsymbol{u} \in V^N : d_{\mathrm{H},l_1+1,l_2}(\boldsymbol{u},\boldsymbol{v}) \ge 1, \\ &p_{l_2+1,l}(\boldsymbol{u}) = p_{l_2+1,l}(\boldsymbol{v}(2^{l-1}+2^{l_2})), d_{\mathrm{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]. \end{aligned} \tag{2.41}$$

is a $(2^{l-1}+2^{l_1}, 2^{l-1}+2^{l_2})$-ruling-out condition.

For l' with $0 \leq l' \leq l-1$ and j' with $2^{l-l'-1} \leq j' < 2^{l-l'}$, we see that an integer i satisfies $j'2^{l'} < i \leq (j'+1)2^{l'}$ if and only if

$$p_{l'+1,N}(e(i)) = p_{l'+1,N}(e(j'2^{l'}+1)).$$

Hence, from (2.24) and (2.25) we see also that

$$R_p((j'+1)2^{l'}) \setminus R_p(j'2^{l'}) = \{u \in V^N : p_{l'+1,l}(u) = p_{l'+1,l}(v(j'2^{l'}+1)), d_{H,l+1,N}(u,v) = t_0\}. \quad (2.42)$$

If U is a subset of $R_p(j'2^{l'}) \cap C$, from (2.20) and (2.42),

$$L(c_{best}(j'2^{l'})) \leq \underline{L}[\{u \in V^N : p_{l'+1,l}(u) = p_{l'+1,l}(v(j'2^{l'}+1)), d_{H,l+1,N}(u,v) = t_0\} \cap V(U)] \quad (2.43)$$

is a $(j'2^{l'}, (j'+1)2^{l'})$-ruling-out condition. If $l' = 0$, then $2^{l-1} \leq j' < 2^l$ and (2.43) gives a $(j', j'+1)$-ruling-out condition:

$$L(c_{best}(j')) \leq \underline{L}[\{u \in V^N : p_{1,l}(u) = p_{1,l}(v(j'+1)), d_{H,l+1,N}(u,v) = t_0\} \cap V(U)] \quad (2.44)$$

If $l' = l-1$, then $j' = 1$ and (2.43) gives a $(2^{l-1}, 2^l)$-ruling-out condition:

$$L(c_{best}(2^{l-1})) \leq \underline{L}[\{u \in V^N : u_l \neq v_l, d_{H,l+1,N}(u,v) = t_0\} \cap V(U)]. \quad (2.45)$$

Since we have

$$\begin{aligned} R_p(2^\tau) \setminus R_p(2^{l-1}) &= \bigcup_{l \leq i \leq \tau} \left(R_p(2^i) \setminus R_p(2^{i-1})\right) \\ &= \bigcup_{l \leq i \leq \tau} \{u \in V^N : u_i \neq v_i, d_{H,i+1,N}(u,v) = t_0\}, \end{aligned}$$

if U is a subset of $R_p(2^{l-1}) \cap C$, then from (2.20),

$$L(c_{best}(2^{l-1})) \leq \min_{l \leq i \leq \tau} \{\underline{L}[\{u \in V^N : u_i \neq v_i, d_{H,i+1,N}(u,v) = t_0\} \cap V(U)]\} \quad (2.46)$$

is a Cond_{ET} at the 2^{l-1}-th iterative step.

Clearly, for any $i \leq j < j'$ and subset U of $R_p(j) \cap C$, the following

$$L(c_{best}(j)) \leq \underline{L}[(R_p(j') \setminus R_p(i)) \cap V(U)] \quad (2.47)$$

is also a (j, j')-ruling-out condition. According to (2.47), we can also get some useful Cond_{RO}'s further.

For example, for $2^{l-1}+2^{l_1} \le j < 2^{l-1}+2^{l_2}$ with $0 \le l_1 < l_2 \le l-1$ and subset U of $R_p(j) \cap C$,

$$\begin{aligned} L(\boldsymbol{c}_{\text{best}}(j)) &\le \underline{L}[(R_p(2^{l-1}+2^{l_2}) \setminus R_p(2^{l-1}+2^{l_1})) \cap V(U)] \\ &= \underline{L}[\{\boldsymbol{u} \in V^N : d_{\text{H},l_1+1,l_2}(\boldsymbol{u},\boldsymbol{v}) \ge 1, p_{l_2+1,l}(\boldsymbol{u}) = p_{l_2+1,l}(\boldsymbol{v}(2^{l-1}+2^{l_2}), \\ &\qquad d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)]. \end{aligned} \quad (2.48)$$

is a $(j, 2^{l-1}+2^{l_2})$-ruling-out condition.

For $j'2^{l'} \le j < (j'+1)2^{l'}$ with $0 \le l' \le l-1$ and $2^{l-l'-1} \le j' < 2^{l-l'}$, and subset U of $R_p(j) \cap C$,

$$\begin{aligned} L(\boldsymbol{c}_{\text{best}}(j)) &\le \underline{L}[(R_p((j'+1)2^{l'}) \setminus R_p(j'2^{l'})) \cap V(U)] \\ &= \underline{L}[\{\boldsymbol{u} \in V^N : p_{l'+1,l}(\boldsymbol{u}) = p_{l'+1,l}(\boldsymbol{v}((j'+1)2^{l'})), \\ &\qquad d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)] \end{aligned} \quad (2.49)$$

is a $(j, (j'+1)2^{l'})$-ruling-out condition.

For $2^{l-1} \le j < 2^l$ and subset U of $R_p(j) \cap C$,

$$\begin{aligned} L(\boldsymbol{c}_{\text{best}}(j)) &\le \underline{L}[(R_p(2^l) \setminus R_p(2^{l-1})) \cap V(U)] \\ &= \underline{L}[\{\boldsymbol{u} \in V^N : u_l \ne v_l, d_{\text{H},l+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_0\} \cap V(U)] \end{aligned} \quad (2.50)$$

is a $(j, 2^l)$-ruling-out condition.

2.4.3 The Use of the Testing Conditions

In Subsections 2.4.1 and 2.4.2, we present a few testing conditions for the Chase-type decoding algorithm. Below we give some suggestions about how to use these conditions factually.

A. At the 2^{l-1}-th iterative step, (2.30) is tested. For $2^{l-1} < j < 2^l$, a Cond_{opt} is tested if and only if the best candidate codeword is renewed, i.e. $\boldsymbol{c}_{\text{best}}(j) \ne \boldsymbol{c}_{\text{best}}(j-1)$. A favorable order of the Cond_{opt}'s is as follows: a) (2.39), b) (2.38), c) (2.35), d) (2.34), e) (2.32).

B. Since the set U in (2.37) can be chosen as a subset of $R_p(i) \cap C$, the strongest

Cond_{opt} that has been evaluated before the j-th step is also a sufficient condition of the optimality of $c_{\text{best}}(j)$, and it is recommended to test this one before any other one is tested.

C. At the 2^{l-1}-th iterative step, it is suggested to find an integer l_1 with $0 \le l_1 \le l-1$ as large as possible such that (2.43) holds with $l' = l_1$ and $j' = 2^{l-l_1-1}$ firstly. If it is failure in finding such an integer l_1, the bounded distance-t_0 decoding is implemented around $v(2^{l-1}+1)$ and let $l_1 = 0$. Then it is suggested to find an integer l_2 with $l_1 < l_2 \le l-1$ as large as possible such that (2.41) holds. If it is failure in finding such an integer l_2, the successive iterative steps are implemented till the best codeword candidate is renewed, otherwise, it is suggested to find an integer l_1 with $0 \le l_1 \le l_2$ as large as possible such that (2.43) holds with $l' = l_1$ and $j' = 2^{l-l_1-1}+2^{l_2-l_1}$, and so on. When the best codeword candidate is renewed, its optimality is tested. If no optimality condition is satisfied, the condition (2.50) is tested. If (2.50) is false, it is suggested to find an integer l_2 as large as possible such that (2.48) holds or an integer l' as large as possible such that (2.49) holds. If it is failure in finding such integers, then the bounded distance-t_0 decoding is implemented till the best codeword candidate is renewed, and so on.

D. Since in (2.47) the set U can be a subset of $R_p(i) \cap C$, the maximum of all the (i, j')-ruling-out conditions with $i \le j$ that have been evaluated before the j-th iterative step is also an (j, j')-ruling-out condition, and it is recommended to test this one before any other (i, j')-ruling-out condition is tested.

E. About the choice of the subset U of $R_p(j) \cap C$. Since the computational complexity for testing of the condition grows when $|U|$ increases, U is usually chosen as a small subset of $R_p(j) \cap C$, for example, let U consist of: a) the best candidate codeword $c_{\text{best}}(j)$, b) the best two candidate codewords, c) the best candidate codeword $c_{\text{best}}(j)$ and the newest candidate codeword, etc. Sometimes, the testing conditions with $U = \phi$ also works well.

F. We also note that if $d_{\text{H}}(v(j), c) \le t_0$ for some codeword c then the candidate codeword generated at the j-th iterative step must be c, i.e. $c(j) = c$. Hence, at the j-th iterative step the bounded distance-t_0 decoding is need to be implemented only if

$$d_{\mathrm{H}}(\boldsymbol{v}(j), \boldsymbol{c}) \geq t_0 + 1, \text{ for all } \boldsymbol{c} \in R_{\mathrm{p}}(j) \cap C. \tag{2.51}$$

□

In [11] and [12], Kaneko et al. proposed two IDAs by considering the Chase($\boldsymbol{z}, t_0, \tau$) combinating with some testing conditions, where $t_0 \triangleq \lfloor (d_{\min} - 1)/2 \rfloor$ and $\tau \triangleq N - t_0$. Since $t_0 + \tau = N$, the Chase($\boldsymbol{z}, \tau, t_0$) is MLD in this case. Below we give a review to these two IDAs. The testing conditions used in [11] and [12] are some sub-cases of those proposed in the previous section. Some suggestions for improving the testing conditions without big increase in the computational complexity are also proposed.

For any two codewords $\boldsymbol{c}'$ and $\boldsymbol{c}''$ of C, let

$$L_1(\boldsymbol{c}', \boldsymbol{c}'') \triangleq \sum_{i \in \varpi_1(\boldsymbol{c}', \boldsymbol{c}'')} |r_i|, \tag{2.52}$$

where

$$\varpi_1(\boldsymbol{c}', \boldsymbol{c}'') \triangleq \mathcal{D}_0(\boldsymbol{c}')^{(d_{\min} - \lfloor (|\mathcal{D}_1(\boldsymbol{c}')| + |\mathcal{D}_1(\boldsymbol{c}'')|)/2 \rfloor)}. \tag{2.53}$$

For any two codewords $\boldsymbol{c}'$ and $\boldsymbol{c}''$ of C and an integer l with $1 \leq l < \tau$, let

$$L_2(\boldsymbol{c}', \boldsymbol{c}'', l) \triangleq \sum_{i \in \varpi_2(\boldsymbol{c}', \boldsymbol{c}'', l)} |r_i|, \tag{2.54}$$

where

$$\varpi_2(\boldsymbol{c}', \boldsymbol{c}'', l) \triangleq \mathcal{D}_0(\boldsymbol{c}')^{(d_{\min} - \lfloor (|\mathcal{D}_1(\boldsymbol{c}')| + |\mathcal{D}_1(\boldsymbol{c}'')|)/2 \rfloor - t_0 - 1)} \cup \{l+1, l+2, \ldots, l+t_0+1\}. \tag{2.55}$$

By use of

$$|\mathcal{D}_1(\boldsymbol{u}) \cap \mathcal{D}_0(\boldsymbol{c}')| \geq d_{\min} - \left\lfloor \frac{|\mathcal{D}_1(\boldsymbol{c}')| + |\mathcal{D}_1(\boldsymbol{c}'')|}{2} \right\rfloor, \text{ for } \boldsymbol{u} \in V^N_{d_{\min}, d_{\min}}(\boldsymbol{c}', \boldsymbol{c}''),$$
$$|\mathcal{D}_1(\boldsymbol{u}) \cap [l+1, N]| \geq t_0 + 1, \text{ for } \boldsymbol{u} \in R_{\mathrm{p}}(2^{\tau}) \setminus R_{\mathrm{p}}(2^{l}),$$

the following two inequalities are given in [12]:

$$L(\boldsymbol{u}) \geq L_1(\boldsymbol{c}', \boldsymbol{c}''), \text{ for } \boldsymbol{u} \in V^N_{d_{\min}, d_{\min}}(\boldsymbol{c}', \boldsymbol{c}''),$$
$$L(\boldsymbol{u}) \geq L_2(\boldsymbol{c}', \boldsymbol{c}'', l), \text{ for } \boldsymbol{u} \in (R_{\mathrm{p}}(2^{\tau}) \setminus R_{\mathrm{p}}(2^{l})) \cap V^N_{d_{\min}, d_{\min}}(\boldsymbol{c}', \boldsymbol{c}'').$$

For the IDA proposed in [12], after the generation of a new candidate $c_{\text{best}}(j)$, i.e. $c_{\text{best}}(j) \neq c_{\text{best}}(j-1)$, the condition

$$L(c_{\text{best}}(j)) \leq L_1(c_{\text{best}}(j), c(1)) \tag{2.56}$$

is tested as an optimality condition of $c(j)$, and an integer T is chosen as the smallest one which satisfies

$$L(c_{\text{best}}(j)) \leq L_2(c_{\text{best}}(j), c(1), T), \tag{2.57}$$

and then it is concluded that the decoding process can be terminated after finishing the 2^T-th iterative step, or in other words,

$$L(c_{\text{best}}(j)) \leq L_2(c_{\text{best}}(j), c(1), l), \tag{2.58}$$

is chosen as a Cond_{ET} at the 2^l-th iterative step.

We notice that the computational complexity of the IDA can be reduced simply by replacing the determination of T after the generation of each new candidate $c_{\text{best}}(j)$ with the test of the Cond_{ET} (2.58) for $j = 2^l$ at the end of the 2^l-th iterative step. We also see easily that the testing conditions (2.56) and (2.58) will be more powerful if the $c(1)$ is replaced simply by a candidate $c(i_0)$ which satisfies $|\mathcal{D}_1(c(i_0))| = \min_{1 \leq i \leq j} |\mathcal{D}_1(c(i))|$, and the computational complexity is almost the same.

Clearly, the following Cond_{opt} given by Lemma 2.1:

$$L(c_{\text{best}}(j)) \leq \underline{L}[V^N_{d_{\min}, d_{\min}}(c_{\text{best}}(j), c(1))], \tag{2.59}$$

is stronger than the Cond_{opt} (2.56), and the following Cond_{ET} given by (2.46):

$$L(c_{\text{best}}(j)) \leq \underline{L}[(R_p(2^T) \setminus R_p(2^l)) \cap V^N_{d_{\min}, d_{\min}}(c_{\text{best}}(j), c(1))], \tag{2.60}$$

is stronger than the Cond_{ET} (2.58). The information of $c(1)$ used in the testing conditions (2.56) and (2.58) is the cardinality $|\mathcal{D}_1(c(1))|$ of the set $\mathcal{D}_1(c(1))$ only. This is one of their advantages, and a reason for less powerful as well. According to the result of Section 4.3, we have

$$\underline{L}[V^N_{d_{\min}, d_{\min}}(c_{\text{best}}(j), c(1))] = \sum_{i \in \varpi_3} |r_i|,$$

where

$$\varpi_3 \triangleq \begin{cases} (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor \zeta/2 \rfloor)})^{(d_{\min} - |\mathcal{D}_1(\mathbf{c}_{\text{best}}(j))|)}, & \text{if } \zeta \geq 0, \\ (\mathcal{D}_{00} \cup \mathcal{D}_{10}^{(\lfloor (-\zeta)/2 \rfloor)})^{(d_{\min} - |\mathcal{D}_1(\mathbf{c}(1))|)}, & \text{if } \zeta < 0, \end{cases}$$

$$\mathcal{D}_{00} \triangleq \mathcal{D}_0(\mathbf{c}_{\text{best}}(j)) \cap \mathcal{D}_0(\mathbf{c}(1)),$$

$$\mathcal{D}_{01} \triangleq \mathcal{D}_0(\mathbf{c}_{\text{best}}(j)) \cap \mathcal{D}_1(\mathbf{c}(1)),$$

$$\mathcal{D}_{10} \triangleq \mathcal{D}_1(\mathbf{c}_{\text{best}}(j)) \cap \mathcal{D}_0(\mathbf{c}(1)),$$

$$\zeta \triangleq |\mathcal{D}_1(\mathbf{c}(1))| - |\mathcal{D}_1(\mathbf{c}_{\text{best}}(j))|.$$

We see that the evaluation of $\underline{L}[V^N_{d_{\min},d_{\min}}(\mathbf{c}_{\text{best}}(j), \mathbf{c}(1))]$ is almost as easy as that of $L_1(\mathbf{c}_{\text{best}}(j), \mathbf{c}(1))$. In general, the computational complexity for evaluating the $\underline{L}[(R_p(2^\tau) \setminus R_p(2^l)) \cap V^N_{d_{\min},d_{\min}}(\mathbf{c}_{\text{best}}(j), \mathbf{c}(1))]$ is greater than $L_2(\mathbf{c}_{\text{best}}(j), \mathbf{c}(1), l)$ in some extent. However, for the point of view of saving the number of bounded distance decodings, it is worthwhile to test the more powerful testing conditions if only the computational complexity for evaluating testing conditions is not greater significantly. We also see that the testing conditions (2.59) and (2.60) will be more powerful if the $\mathbf{c}(1)$ is replaced simply by a suitably selected candidate, such as the second best one among the candidates which have been obtained so far, and the computational complexity also remains to be the same by and large.

Now we consider the testing conditions used in [11]. For $2^{l-1} \leq j < 2^l$, let l' be the largest integer such that $2^{l'}|j$ and let $j' = j2^{-l'}$, by taking $U = \phi$ in (2.43) we get a $(j, j + 2^{l'})$-ruling-out condition as the following:

$$\begin{aligned} L(\mathbf{c}_{\text{best}}(j)) &\leq \underline{L}[\{\mathbf{u} \in V^N : p_{l'+1,l}(\mathbf{u}) = p_{l'+1,l}(\mathbf{v}(j+1)), d_{\text{H},l+1,N}(\mathbf{u}, \mathbf{z}) = t_0\}] \\ &= \sum_{i=l'+1}^{l-1} e(j+1)_i |r_i| + \sum_{i=0}^{t_0} |r_{i+l}|. \end{aligned} \tag{2.61}$$

If $j = 2^{l-1}$, then (2.61) becomes

$$L(\mathbf{c}_{\text{best}}(2^{l-1})) \leq \underline{L}[\{\mathbf{u} \in V^N : u_l \neq z_l, d_{\text{H},l+1,N}(\mathbf{u}, \mathbf{z}) = t_0\}] = \sum_{i=0}^{t_0} |r_{i+l}|, \tag{2.62}$$

it is a $(2^{l-1}, 2^l)$-ruling-out condition. If (2.62) holds for $l = l_0$, then $\mathbf{c}_{\text{best}}(2^{l_0-1}) = \mathbf{c}_{\text{best}}(2^{l_0})$ and thus from $|r_{l_0}| \leq |r_{t_0+l_0+1}|$ we see that (2.62) holds for $l = l_0 + 1$. Hence, (2.62) is a Cond_{ET} at the end of the 2^{l-1}-th iterative step. Because the

IDA is MLD, (2.62) is also a Cond_{opt} for $c_{\text{best}}(2^{l-1})$. Indeed, this can also be concluded simply from (2.30) with $U = \phi$.

For the IDA proposed in [11], the ruling-out condition (2.61) is tested before each iterative step. If the next test error pattern is $e(2^{l-1}+1)$ and (2.61) is satisfied, then the $c_{\text{best}}(2^{l-1})$ is output as the decoded codeword (it is the optimal codeword) and the decoding process is terminated.

2.5 Testing Conditions for Generalizations of Chase Algorithm

In [27], an IDA is proposed to generalize the Chase algorithm by shifting the way of generating the search centers around which the bounded distance-t_0 decoding is performed. More precisely, a series of words $v(j) \in V^N$ are determined as the search centers before the decoding, such as the original Chase algorithm, or during the decoding process. In the j-th iterative step, the bounded distance-t_0 decoding is performed around $v(j)$. Let $c(j)$ be the output of the j-th iterative step and write

$$u(j) \triangleq \begin{cases} c(j), & \text{if } c(j) \neq *, \\ v(j), & \text{if } c(j) = *, \end{cases} \tag{2.63}$$

$$d(j) \triangleq \begin{cases} d_{\min}, & \text{if } c(j) \neq *, \\ t_0 + 1, & \text{if } c(j) = *. \end{cases} \tag{2.64}$$

Then we have

$$R_p(j) \supseteq \bigcup_{i=1}^{j} O_{d(i)-1}(u(i)).$$

Thus from $L(c_{\text{best}}(j)) = \underline{L}[R_p(j)]$ and lemma 2.1,

$$L(c_{\text{best}}(j)) \leq \underline{L}[V^N_{d(1),d(2),\ldots,d(j)}(u(1), u(2), \ldots, u(j))] \tag{2.65}$$

is an optimality condition of $c_{\text{best}}(j)$.

About the determination of the search centers, the following method is proposed in [17]: At the end of the j-th iterative step, the next search center $v(j+1)$ is chosen in $V^N_{d_1,d_2,\ldots,d_h}(u_1, u_2, \ldots, u_h)$ such that

$$L(v(j+1)) = \underline{L}[V^N_{d_1,d_2,\ldots,d_h}(u_1, u_2, \ldots, u_h)], \tag{2.66}$$

where the pairs $(\boldsymbol{u}_1, d_1)$, $(\boldsymbol{u}_2, d_2)$, ..., $(\boldsymbol{u}_h, d_h)$ are selected from the pairs $(\boldsymbol{u}(1), d(1))$, $(\boldsymbol{u}(2), d(2))$,...,$(\boldsymbol{u}(j), d(j))$ which are defined by (2.63) and (2.64).

2.6 Testing Conditions for Iterative MWTS Decoding Algorithm

In the iterative minimum-weight sub-trellis search (MWTS) decoding algorithm [19] for binary linear codes, the first codeword candidate $\boldsymbol{c}(1)$ is obtained by a simple decoder, such as the ordered statistics decoding of the zero-th order or the first order [3]. At the j-th iterative step with $j > 1$, the next candidate codeword $\boldsymbol{c}(j)$ is the best codeword in $\{\boldsymbol{v} \in C : d_{\mathrm{H}}(\boldsymbol{v}, \boldsymbol{c}(j-1)) = w_1\}$ which is obtained by MWTS around $\boldsymbol{c}(j-1)$, where $w_1 (= d_{\min})$ is the minimum weight. There is no decoding failure at each iterative step.

Then for $j > 1$ the search region at the j-th iterative step is

$$R(j) = O_{w_2-1}(\boldsymbol{c}(j-1)) \setminus \{\boldsymbol{c}(j-1)\},$$

where w_2 is the second smallest weight of C. Since $\boldsymbol{c}(i) \in R(i)$ for $i \geq 1$, we have that

$$O_{w_1-1}(\boldsymbol{c}(i)) \subseteq R(i), \text{ for } i \geq 1,$$

and thus, for $j > 1$,

$$\begin{aligned} R_{\mathrm{p}}(j) &= R_{\mathrm{p}}(j) \cup \left(\bigcup_{i=1}^{j} O_{w_1-1}(\boldsymbol{c}(i)) \right) \\ &\supseteq \left(\bigcup_{i=2}^{j} (O_{w_2-1}(\boldsymbol{c}(i-1)) \setminus \{\boldsymbol{c}(i-1)\}) \right) \cup \left(\bigcup_{i=1}^{j} O_{w_1-1}(\boldsymbol{c}(i)) \right) \\ &= \left(\bigcup_{i=1}^{j-1} O_{w_2-1}(\boldsymbol{c}(i)) \right) \cup O_{w_1-1}(\boldsymbol{c}(j)). \end{aligned}$$

Hence, from $L(\boldsymbol{c}_{\mathrm{best}}(j)) = \underline{L}[R_{\mathrm{p}}(j)]$ and lemma 2.1,

$$L(\boldsymbol{c}_{\mathrm{best}}(j)) \leq \underline{L}[V^N_{w_2,\ldots,w_2,w_1}(\boldsymbol{c}(1), \ldots, \boldsymbol{c}(j-1), \boldsymbol{c}(j))] \tag{2.67}$$

is an optimality condition of $\boldsymbol{c}_{\mathrm{best}}(j)$.

2.7 Testing Conditions for Generalizations of GMD Decoding

2.7.1 GMD-Like Decoding

For nonnegative integers s and t such that $s + 2t < d_{\min}$ and $\boldsymbol{v} \in V^N$, the decoding which corrects s erasures in the first s bit positions and t or less errors in the remaining bit positions of input $\boldsymbol{v}$ is called **(s,t)-decoding with respect to $\boldsymbol{v}$.** Let

$$p \triangleq \begin{cases} 0, & \text{if } d_{\min} \text{ is even,} \\ 1, & \text{otherwise.} \end{cases} \tag{2.68}$$

For $\boldsymbol{v} \in V^N$, the **GMD-like decoding GMD($\boldsymbol{v}$)** is defined as an IDA for which the $(2j-p-1, \rho-j)$-decoding with respect to $\boldsymbol{v}$ is executed in the j-th iterative step for $1 \le j \le \rho \triangleq (d_{\min}+p)/2$. GMD($\boldsymbol{z}$) is the original GMD proposed by Forney [2]. For GMD($\boldsymbol{v}$), we have

$$R(j) = \{\boldsymbol{u} \in V^N : d_{\mathrm{H},2j-p,N}(\boldsymbol{u},\boldsymbol{v}) \le \rho - j\}, \text{ for } 1 \le j \le \rho. \tag{2.69}$$

For simplicity, write

$$T(i,d,\boldsymbol{v}) \triangleq \{\boldsymbol{u} \in V^N : d_{\mathrm{H},2i-p,N}(\boldsymbol{u},\boldsymbol{v}) \ge d\}. \tag{2.70}$$

It follows from (2.69) that

$$L(\boldsymbol{c}_{\text{best}}(j)) \le \underline{L}\left[\bigcap_{i=1}^{j} T(i,\rho-i+1,\boldsymbol{v})\right] \tag{2.71}$$

is an optimality condition of $\boldsymbol{c}_{\text{best}}(j)$ for $1 \le j \le \rho$.

It is proved in [18] that the N-tuple $\boldsymbol{u}^{(j)} = (u_1^{(j)}, u_2^{(j)}, \ldots, u_N^{(j)})$ with

$$u_i^{(j)} = \begin{cases} z_i + 1, & \text{if } i+p \text{ is even and } 1 \le (i+p)/2 < j, \\ & \qquad\qquad \text{or } 2j-p \le i \le \rho+j-p, \\ z_i, & \text{otherwise,} \end{cases} \tag{2.72}$$

is the tuple in $\bigcap_{i=1}^{j} T(i,\rho-i+1,\boldsymbol{z})$ such that

$$L(\boldsymbol{u}^{(j)}) = \underline{L}\left[\bigcap_{i=1}^{j} T(i,\rho-i+1,\boldsymbol{z})\right]. \tag{2.73}$$

In the remaining part of this section, let j be a fixed integer satisfying $1 \leq j \leq \rho$. Below for any N-tuple $\boldsymbol{v} \in V^N$, with a method similar to that used in [18], we will determine a tuple $\boldsymbol{u}^*$ in the set

$$\mathcal{T}_j(\boldsymbol{v}) \triangleq \bigcap_{i=1}^{j} T(i, \rho - i + 1, \boldsymbol{v}) \tag{2.74}$$

such that

$$L(\boldsymbol{u}^*) = \underline{L}[\mathcal{T}_j(\boldsymbol{v})]. \tag{2.75}$$

For any N-tuple $\boldsymbol{u} \in V^N$ and integers i' and i'' with $1 \leq i' \leq i'' \leq N$, define

$$L_{i',i''}(\boldsymbol{u}) \triangleq \sum_{i' \leq i \leq i'', u_i \neq z_i} |r_i|. \tag{2.76}$$

According the result of Section 4.3, we have

$$\underline{L_{2j-p,N}}[T(j,d,\boldsymbol{v})] = \sum_{i \in (\mathcal{D}_0(\boldsymbol{v}) \cap [2j-p,N])^{(d-\tau)}} |r_i|, \tag{2.77}$$

where

$$\tau \triangleq |\mathcal{D}_1(\boldsymbol{v}) \cap [2j - p, N]|. \tag{2.78}$$

Clearly,

$$\underline{L_{2j-p,N}}[T(j,d,\boldsymbol{v})] = \begin{cases} 0, & \text{for } d \leq \tau, \\ +\infty, & \text{for } d > N - 2j + p + 1, \end{cases} \tag{2.79}$$

and for $\tau \leq d \leq N - 2j + p$,

$$\underline{L_{2j-p,N}}[T(j,d+1,\boldsymbol{v})] - \underline{L_{2j-p,N}}[T(j,d,\boldsymbol{v})] \geq |r_{2j-p}|. \tag{2.80}$$

We define integers f_j, f_{j-1}, $\ldots$, f_1 by induction as the followings:

$$f_j \triangleq \begin{cases} \tau, & \text{if } \rho - j + 1 \leq \tau, \\ \rho - j + 1, & \text{if } \rho - j + 1 > \tau, \end{cases} \tag{2.81}$$

and for $1 \leq i < j$,

$$f_i \triangleq \begin{cases} f_{i+1} + |\mathcal{D}_1(\boldsymbol{v}) \cap \{2i - p, 2i - p + 1\}|, \text{ if } f_{i+1} > \rho - i, \\ f_{i+1} + 2, \text{ if } f_{i+1} \leq \rho - i \text{ and } v_{2i-p} = v_{2i-p+1} = 1, \\ f_{i+1} + 1, \text{ otherwise.} \end{cases} \tag{2.82}$$

We can see easily that

$$f_i \geq \rho - i + 1, \text{ for } i = 1, 2, \ldots, j. \tag{2.83}$$

Let $\boldsymbol{u}^*$ denote the N-tuple in V^N such that

$$\mathcal{D}_1(\boldsymbol{u}^*) \cap [2j - p, N] = (\mathcal{D}_0(\boldsymbol{v}) \cap [2j - p, N])^{(\rho - j + 1 - \tau)}, \tag{2.84}$$

and for $1 \leq i < 2j - p$,

$$u_i^* = \begin{cases} z_i + 1, & \text{if } i + p \text{ is even, } f_{(i+p)/2+1} = \rho - (i + p)/2, \\ & \qquad v_i = z_i \text{ and } v_{i+1} = z_{i+1}, \\ z_i, & \text{otherwise.} \end{cases} \tag{2.85}$$

From the definitions, we can see easily that

$$f_i = d_{\mathrm{H}, 2i-p, N}(\boldsymbol{u}^*, \boldsymbol{v}), \text{ for } i = 1, 2, \ldots, j, \tag{2.86}$$

and thus by (2.83) we see that $\boldsymbol{u}^*$ belongs to $\mathcal{T}_j(\boldsymbol{v})$. We have the following lemma.

Lemma 2.5: $\boldsymbol{u}^*$ belongs to $\mathcal{T}_j(\boldsymbol{v})$ and satisfies (2.75).

Proof: See Appendix A. □

Remark: If $\boldsymbol{v} = \boldsymbol{z}$, then $\tau = 0$ and $f_i = \rho - i + 1$ for $1 \leq i \leq j$ and thus $\boldsymbol{u}^*$ is just the tuple given by (2.72).

2.7.2 Multistage GMD-Like Decoding

For a positive integer h, a **multistage GMD-like decoding** [13] consists of successive GMD($\boldsymbol{v}^{(l)}$) for $1 \leq l \leq h$, where $\boldsymbol{v}^{(1)} \triangleq \boldsymbol{z}$ and $\boldsymbol{v}^{(l)}$ is the best word in $\bigcap_{l'=1}^{l-1} \mathcal{T}_\rho(\boldsymbol{v}^{(l')})$ which satisfies

$$L(\boldsymbol{v}^{(l)}) = \underline{L}\left[\bigcap_{l'=1}^{l-1} \mathcal{T}_\rho(\boldsymbol{v}^{(l')})\right], \text{ for } 1 < l \leq h. \tag{2.87}$$

From (2.72), we see that $\boldsymbol{v}^{(2)} = (v_1^{(2)}, v_2^{(2)}, \ldots, v_N^{(2)})$ is given by

$$v_i^{(2)} = \begin{cases} z_i + 1, & \text{if } i + p \text{ is even and } 1 \leq (i + p)/2 \leq \rho, \\ z_i, & \text{otherwise.} \end{cases} \tag{2.88}$$

Let $\boldsymbol{u}^{(l)}_{\text{best}}(j)$ denote the best codeword candidate at the end of the j-th iterative step of GMD($\boldsymbol{v}^{(l)}$). Then the following

$$L(\boldsymbol{u}^{(l)}_{\text{best}}(j)) \le \underline{L}\left[\mathcal{T}_j(\boldsymbol{v}^{(l)}) \cap \left(\bigcap_{l'=1}^{l-1} \mathcal{T}_\rho(\boldsymbol{v}^{(l')})\right)\right], \tag{2.89}$$

is an optimality condition of $\boldsymbol{u}^{(l)}_{\text{best}}(j)$. Below we consider to evaluate right-hand of (2.89) for $l = 2$ and $1 \le j \le \rho$.

Let j be a fixed integer with $1 \le j \le \rho$. Let U^* be the set of tuples $\boldsymbol{u}^*$ in $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ which satisfy

$$L(\boldsymbol{u}^*) = \underline{L}[\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})]. \tag{2.90}$$

Thus the right-hand side of (2.90) can be evaluated by any tuple in U^*. To give such a tuple, we give a lemma at first.

Lemma 2.6: There is a tuple $\boldsymbol{u}^*$ in U^* such that
1) $d_{\text{H},2\rho-p,N}(\boldsymbol{u}^*, \boldsymbol{z}) = 1$.
2) For $i \in [1, N] \setminus [2-p, 2\rho - p + 1]$, $u^*_i = z_i$.
3) For $1 \le i \le \rho$,

$$\{u^*_{2i-p} + z_{2i-p}, u^*_{2i-p+1} + z_{2i-p+1}\} = V. \tag{2.91}$$

Proof: Let $\boldsymbol{u}^*$ be an arbitrary tuple in U^*. Clearly, we have $d_{\text{H},2\rho-p,N}(\boldsymbol{u}^*, \boldsymbol{z}) \ge 1$. If $d_{\text{H},2\rho-p,N}(\boldsymbol{u}^*, \boldsymbol{z}) \ge 2$, we will prove that there is a tuple $\boldsymbol{u}'$ in U^* such that

$$d_{\text{H},2\rho-p,N}(\boldsymbol{u}', \boldsymbol{z}) = d_{\text{H},2\rho-p,N}(\boldsymbol{u}^*, \boldsymbol{z}) - 1.$$

Let i' denote the smallest integer with $i' \ge 2\rho - p$ and $u^*_{i'} \ne z_{i'}$. If $u^*_{2i-p+1} \ne z_{2i-p+1}$ for all $1 \le i < \rho$, then the tuple $\boldsymbol{u}'$ can be chosen as the tuple such that

$$u'_{i'} = z_{i'} \text{ and } u'_i = u^*_i, \text{ for } i \ne i'.$$

If $u^*_{2i-p+1} = z_{2i-p+1}$ for some integer i with $1 \le i < \rho$, let i'' be the largest one of such kind of integers. Then the tuple $\boldsymbol{u}'$ can be chosen as the tuple such that

$$u'_{i'} = z_{i'},\ u'_{2i''-p+1} \ne z_{2i''-p+1},$$
$$u'_i = u^*_i, \text{ for } i \in [1, N] \setminus \{i', 2i'' - p + 1\}.$$

By induction, we can assume that $d_{\mathrm{H},2\rho-p,N}(\boldsymbol{u}^*, \boldsymbol{z}) = 1$, i.e. $\boldsymbol{u}^*$ satisfies 1).

If $u^*_{i'} \neq z_{i'}$ for some i' with $i' \geq 2\rho + 2 - p$, then the tuple $\boldsymbol{u}'$ with

$$\begin{aligned} &u'_{i'} = z_{i'},\ u'_{2\rho-p+1} \neq z_{2\rho-p+1}, \\ &u'_i = u^*_i,\ \text{ for } i \in [1, N] \setminus \{i', 2\rho - p + 1\}, \end{aligned}$$

is also in $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ and $L(\boldsymbol{u}') \leq L(\boldsymbol{u}^*)$.

If $p = 0$ and $u^*_1 \neq z_1$, then the tuple $\boldsymbol{u}'$ with

$$u'_1 = z_1, \text{ and } u'_i = u^*_i, \text{ for } i > 1,$$

is also in $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ and $L(\boldsymbol{u}') \leq L(\boldsymbol{u}^*)$.

Thus, we can also assume that $u^*_i = z_i$ for $i \in [1, N] \setminus [2 - p, 2\rho - p + 1]$, i.e. 2) holds.

Clearly, (2.91) holds for $i = \rho$.

If there is an integer i with $1 \leq i < \rho$ such that

$$u^*_{2i-p} = z_{2i-p} \text{ and } u^*_{2i-p+1} = z_{2i-p+1},$$

let i' be the largest one of such kind of integers. Then there must exist an integer i'' with $i' < i'' < \rho$ such that

$$u^*_{2i''-p} \neq z_{2i''-p} \text{ and } u^*_{2i''-p+1} \neq z_{2i''-p+1}.$$

We can see easily that the tuple $\boldsymbol{u}'$ with

$$\begin{aligned} &u'_{2i'-p+1} \neq z_{2i'-p+1},\ u'_{2i''-p} = z_{2i''-p}, \\ &u'_i = u^*_i,\ \text{ for } i \in [1, N] \setminus \{2i' - p + 1, 2i'' - p\}, \end{aligned}$$

is also in $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ and $L(\boldsymbol{u}') \leq L(\boldsymbol{u}^*)$.

Hence, we can assume further that $1 \in \{u^*_{2i-p} + z_{2i-p}, u^*_{2i-p+1} + z_{2i-p+1}\}$ holds for all $1 \leq i < \rho$.

If $u^*_{2i'-p} \neq z_{2i'-p}$ and $u^*_{2i'-p+1} \neq z_{2i'-p+1}$ for some integer i' with $1 \leq i' < \rho$, then the tuple $\boldsymbol{u}'$ with and

$$u_{2i'-p} = z_{2i'-p}, \text{ and } u_i = u^*_i, \text{ for } i \neq 2i' - p,$$

is in $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ and $L(\boldsymbol{u}') \leq L(\boldsymbol{u}^*)$.

Thus, we have proved that there is a tuple $\boldsymbol{u}^*$ in U^* such that 1), 2) and 3) are valid. □

Let π be a permutation of $[1, \rho]$ such that

$$|r_{2i-p+1}| - |r_{2i-p}| \leq |r_{2i'-p+1}| - |r_{2i'-p}| \tag{2.92}$$

holds for all integers i and i' in $[1, \rho]$ with $\pi(i) < \pi(i')$. Let π^{-1} denote the reverse permutation of π, i.e. π^{-1} is the permutation of $[1, \rho]$ such that

$$\pi^{-1}(\pi(i)) = i, \text{ for } i = 1, 2, \ldots, \rho. \tag{2.93}$$

For any subset I of $[1, \rho]$, write $\pi[I] \triangleq \{\pi(i) : i \in I\}$ and let $\boldsymbol{u}^*(I) \triangleq (u_1^*(I), u_2^*(I), \ldots, u_N^*(I))$ denote the N-tuple in V^N which satisfies

$$u_{2i-p}^*(I) \neq z_{2i-p} \text{ and } u_{2i-p+1}^*(I) = z_{2i-p+1}, \text{ for } i \in I, \tag{2.94}$$

$$u_{2i-p}^*(I) = z_{2i-p} \text{ and } u_{2i-p+1}^*(I) \neq z_{2i-p+1}, \text{ for } i \in [1, \rho] \setminus I, \tag{2.95}$$

$$u_i^*(I) = z_i, \text{ for } i \in [1, N] \setminus [2 - p, 2\rho - p + 1]. \tag{2.96}$$

Define subsets I_j, I_{j-1}, ..., I_1 of $[1, \rho]$ by induction as the following:

$$I_j \triangleq \pi^{-1}[(\pi[[j, \rho]])^{(\lceil(\rho-j+1)/2\rceil)}], \tag{2.97}$$

and for $1 \leq i < j$,

$$I_i \triangleq \begin{cases} I_{i+1} \cup \pi^{-1}[(\pi[[i, \rho] \setminus I_{i+1}])^{(1)}], & \text{if } \rho - i \text{ is even,} \\ I_{i+1}, & \text{otherwise.} \end{cases} \tag{2.98}$$

Then we have the following theorem.

Theorem 2.1: The N-tuple $\boldsymbol{u}^*(I_1)$ is in U^*, i.e. $\boldsymbol{u}^*(I_1)$ belongs to $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ and satisfies

$$L(\boldsymbol{u}^*(I_1)) = \underline{L}[\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})]. \tag{2.99}$$

Remark: $\boldsymbol{v}^{(3)}$ can be chosen as the tuple $\boldsymbol{u}^*(I_1)$ of the case $j = \rho$.

Proof of Theorem 2.1: We will prove that there is a subset $I \subset [1, \rho]$ such that $\boldsymbol{u}^*(I) \in U^*$ and

$$I_i \subseteq I(\boldsymbol{u}^*) \cap [i, \rho] \tag{2.100}$$

holds for $i = 1, 2, \ldots, j$, and thus (2.99) follows from $L(\boldsymbol{u}^*(I_1)) \leq L(\boldsymbol{u}^*(I))$.

It is obvious that, for any $I \subset [1, \rho]$, the tuple $\boldsymbol{u}^*(I)$ belongs to $\mathcal{T}_j(\boldsymbol{v}^{(2)}) \cap \mathcal{T}_\rho(\boldsymbol{z})$ if and only if

$$|I \cap [i, \rho]| \geq \lceil (\rho - i + 1)/2 \rceil, \text{ for } i = 1, 2, \ldots, j. \tag{2.101}$$

Then, from (2.98), we have that

$$|I_1 \cap [i, \rho]| \geq |I_i| = \lceil (\rho - i + 1)/2 \rceil, \text{ for } i = 1, 2, \ldots, j, \tag{2.102}$$

and thus $\boldsymbol{u}^*(I_1) \in \mathcal{T}_j(\boldsymbol{v}^{(2)})$ by (2.101).

Let $\boldsymbol{u}^*$ be an arbitrary tuple in U^* which satisfies 1), 2) and 3) of Lemma 2.6. Let $I(\boldsymbol{u}^*)$ denote the set of integers i with $1 \leq i \leq \rho$ which satisfy

$$u^*_{2i-p} = z_{2i-p} \text{ and } u^*_{2i-p+1} \neq z_{2i-p+1}.$$

Then we have $\boldsymbol{u}^* = \boldsymbol{u}^*(I(\boldsymbol{u}^*))$ and (2.101) holds for $I = I(\boldsymbol{u}^*)$.

If there are two integers i_1 and i_2 in $[j, \rho]$ such that $\pi(i_1) < \pi(i_2)$, $i_1 \notin I(\boldsymbol{u}^*)$ and $i_2 \in I(\boldsymbol{u}^*)$, then (2.101) holds for $I = (I(\boldsymbol{u}^*) \setminus \{i_2\}) \cup \{i_1\}$ and from (2.92) we see that the tuple $\boldsymbol{u}^*((I(\boldsymbol{u}^*) \setminus \{i_2\}) \cup \{i_1\})$ is also in U^*.

Hence, we can assume, without loss of generality, that $\pi(i_1) < \pi(i_2)$ for all $i_1 \in I(\boldsymbol{u}^*) \cap [j, \rho]$ and $i_2 \in [j, \rho] \setminus I(\boldsymbol{u}^*)$. Then, (2.100) holds for $i = j$. Now we assume that (2.100) holds for $i = i_0 + 1$ with $1 \leq i_0 < j$.

If there are two integers i_1 and i_2 in $[i_0, \rho]$ such that $\pi(i_1) < \pi(i_2)$, $i_1 \in [i_0, \rho] \setminus I(\boldsymbol{u}^*)$ and $i_2 \in (I(\boldsymbol{u}^*) \setminus I_{i_0+1}) \cap [i_0, \rho]$, then (2.101) holds for $I = (I(\boldsymbol{u}^*) \setminus \{i_2\}) \cup \{i_1\}$ and from (2.92) we see that the tuple $\boldsymbol{u}^*((I(\boldsymbol{u}^*) \setminus \{i_2\}) \cup \{i_1\})$ is also in U^*.

Hence, we can assume, without loss of generality, that $\pi(i_1) < \pi(i_2)$ for all $i_1 \in (I(\boldsymbol{u}^*) \setminus I_{i_0+1}) \cap [i_0, \rho]$ and $i_2 \in [i_0, \rho] \setminus I(\boldsymbol{u}^*)$. Then (2.100) holds for $i = i_0$.

By induction, we see that there is a tuple $\boldsymbol{u}^*$ in U^* such that (2.100) holds for $i = 1, 2, \ldots, j$. □

Chapter 3

Integer Programming Problem Related to the Testing Conditions for Chase-Type Decoding Algorithm

3.1 The Integer Programming Problem

For most testing conditions given in Chapter 2, it is needed only to evaluate the minimum with the form of

$$\underline{L}[\{\boldsymbol{u} \in V^N : d_{\mathrm{H},l_1+1,l_2}(\boldsymbol{u},\boldsymbol{v}) \geq t_1, p_{l_2+1,l_2'}(\boldsymbol{u}) = p_{l_2+1,l_2'}(\boldsymbol{v}'),$$
$$d_{\mathrm{H},l_2'+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_2\} \cap V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)], \quad (3.1)$$

where $0 \leq l_1 \leq l_2 \leq l_2' \leq N$. Since the partial tuple $p_{l_2+1,l_2'}(\boldsymbol{u})$ is fixed for all $\boldsymbol{u}$ in the set over which the minimum (3.1) is taken, from the sight of evaluating the minimum, we only consider the case $l_2' = l_2$. Thus (3.1) becomes the following

$$\underline{L}[Y(\boldsymbol{v},t_1,l_1,t_2,l_2) \cap V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)], \quad (3.2)$$

where

$$Y(\boldsymbol{v},t_1,l_1,t_2,l_2) \triangleq \{\boldsymbol{u} \in V^N : d_{\mathrm{H},l_1+1,l_2}(\boldsymbol{u},\boldsymbol{v}) \geq t_1, d_{\mathrm{H},l_2+1,N}(\boldsymbol{u},\boldsymbol{v}) = t_2\}. \quad (3.3)$$

Below we present an **integer programming problem** (IPP) whose optimal value of the object function is just equal to the minimum given by (3.2).

Define

$$I_1 \triangleq [1, l_1], \tag{3.4}$$
$$I_2 \triangleq [l_1+1, l_2] \cap \mathcal{D}_0(\boldsymbol{v}), \tag{3.5}$$
$$I_3 \triangleq [l_1+1, l_2] \cap \mathcal{D}_1(\boldsymbol{v}), \tag{3.6}$$
$$I_4 \triangleq [l_2+1, N] \cap \mathcal{D}_0(\boldsymbol{v}), \tag{3.7}$$
$$I_5 \triangleq [l_2+1, N] \cap \mathcal{D}_1(\boldsymbol{v}). \tag{3.8}$$

Then $\{I_1, I_2, I_3, I_4, I_5\}$ is a partition of $[1, N]$. For $\boldsymbol{u} \in V^N$ and $a \in [1, 5]$, let

$$q(\boldsymbol{u})_a \triangleq |\mathcal{D}_1(\boldsymbol{u}) \cap I_a|. \tag{3.9}$$

Then, we have that

$$d_{\mathrm{H}, l_1+1, l_2}(\boldsymbol{u}, \boldsymbol{v}) = |\mathcal{D}_1(\boldsymbol{u}) \cap I_2| + |\mathcal{D}_0(\boldsymbol{u}) \cap I_3| = q(\boldsymbol{u})_2 + |I_3| - q(\boldsymbol{u})_3, \tag{3.10}$$
$$d_{\mathrm{H}, l_2+1, N}(\boldsymbol{u}, \boldsymbol{v}) = |\mathcal{D}_1(\boldsymbol{u}) \cap I_4| + |\mathcal{D}_0(\boldsymbol{u}) \cap I_5| = q(\boldsymbol{u})_4 + |I_5| - q(\boldsymbol{u})_5. \tag{3.11}$$

Hence, $\boldsymbol{u} \in V^N$ is in $Y(\boldsymbol{v}, t_1, l_1, t_2, l_2)$ if and only if

$$\begin{cases} q(\boldsymbol{u})_2 - q(\boldsymbol{u})_3 \geq t_1 - |I_3|, \\ q(\boldsymbol{u})_4 - q(\boldsymbol{u})_5 = t_2 - |I_5|. \end{cases} \tag{3.12}$$

To derive a condition for $\boldsymbol{u} \in V^N$ to be in $V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$, we partition $[1, N]$ again by refining the partition $\{I_1, I_2, I_3, I_4, I_5\}$. For $a \in [1, 5]$ and $\boldsymbol{\sigma} = \sigma_1\sigma_2 \cdots \sigma_h \in B^h$, define

$$\mathcal{D}_{a, \boldsymbol{\sigma}} \triangleq I_a \cap \left(\bigcap_{i=1}^{h} \mathcal{D}_{\sigma_i}(\boldsymbol{u}_i) \right). \tag{3.13}$$

Then, $\{\mathcal{D}_{a, \boldsymbol{\sigma}} : a \in [1, 5] \text{ and } \boldsymbol{\sigma} \in B^h\}$ is a refinement of $\{I_1, I_2, I_3, I_4, I_5\}$ and

$$\mathcal{D}_j(\boldsymbol{u}_i) = \bigcup_{a=1}^{5} \bigcup_{\boldsymbol{\sigma} \in B^h, \sigma_i = j} \mathcal{D}_{a, \boldsymbol{\sigma}}, \text{ for } i \in [1, h] \text{ and } j = 0, 1. \tag{3.14}$$

For $\boldsymbol{u} \in V^N$, $a \in [1, 5]$ and $\boldsymbol{\sigma} \in B^h$, let

$$q(\boldsymbol{u})_{a, \boldsymbol{\sigma}} \triangleq |\mathcal{D}_1(\boldsymbol{u}) \cap \mathcal{D}_{a, \boldsymbol{\sigma}}|. \tag{3.15}$$

Then

$$q(\boldsymbol{u})_a = \sum_{\boldsymbol{\sigma}\in B^h} q(\boldsymbol{u})_{a,\boldsymbol{\sigma}}, \tag{3.16}$$

$$|\mathcal{D}_0(\boldsymbol{u}) \cap \mathcal{D}_{a,\boldsymbol{\sigma}}| = |\mathcal{D}_{a,\boldsymbol{\sigma}}| - q(\boldsymbol{u})_{a,\boldsymbol{\sigma}}. \tag{3.17}$$

Also, from (3.14), (3.15) and (3.17), we have that

$$\begin{aligned} d_{\mathrm{H}}(\boldsymbol{u}, \boldsymbol{u}_i) &= |\mathcal{D}_1(\boldsymbol{u}) \cap \mathcal{D}_0(\boldsymbol{u}_i)| + |\mathcal{D}_0(\boldsymbol{u}) \cap \mathcal{D}_1(\boldsymbol{u}_i)| \\ &= \sum_{a=1}^{5} \left(\sum_{\boldsymbol{\sigma}\in B^h, \sigma_i=0} q(\boldsymbol{u})_{a,\boldsymbol{\sigma}} + \sum_{\boldsymbol{\sigma}\in B^h, \sigma_i=1} (|\mathcal{D}_{a,\boldsymbol{\sigma}}| - q(\boldsymbol{u})_{a,\boldsymbol{\sigma}}) \right) \\ &= \sum_{a=1}^{5} \sum_{\boldsymbol{\sigma}\in B^h} q(\boldsymbol{u})_{a,\boldsymbol{\sigma}} (-1)^{\sigma_i} + |\mathcal{D}_1(\boldsymbol{u}_i)|. \end{aligned} \tag{3.18}$$

Hence, $\boldsymbol{u} \in V^N$ is in $V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$ if and only if

$$\sum_{a=1}^{5} \sum_{\boldsymbol{\sigma}\in B^h} q(\boldsymbol{u})_{a,\boldsymbol{\sigma}} (-1)^{\sigma_i} \geq d_i - |\mathcal{D}_1(\boldsymbol{u}_i)|, \text{ for } i \in [1, h]. \tag{3.19}$$

Let Q denote the set of all the $5 \cdot 2^h$-tuples over nonnegative integers. For $\boldsymbol{q} \in Q$, the $5 \cdot 2^h$ components of $\boldsymbol{q}$ are referred as $q_{a,\boldsymbol{\sigma}}$ with $1 \leq a \leq 5$ and $\boldsymbol{\sigma} \in B^h$. Let Q^* denote the set of all the $5 \cdot 2^h$-tuples $\boldsymbol{q} \in Q$ which satisfy

$$0 \leq q_{a,\boldsymbol{\sigma}} \leq |\mathcal{D}_{a,\boldsymbol{\sigma}}|, \text{ for } a \in [1, 5] \text{ and } \boldsymbol{\sigma} \in B^h, \tag{3.20}$$

$$\sum_{\boldsymbol{\sigma}\in B^h} (q_{2,\boldsymbol{\sigma}} - q_{3,\boldsymbol{\sigma}}) \geq t_1 - |I_3|, \tag{3.21}$$

$$\sum_{\boldsymbol{\sigma}\in B^h} (q_{4,\boldsymbol{\sigma}} - q_{5,\boldsymbol{\sigma}}) = t_2 - |I_5|, \tag{3.22}$$

$$\sum_{a=1}^{5} \sum_{\boldsymbol{\sigma}\in B^h} q_{a,\boldsymbol{\sigma}} (-1)^{\sigma_i} \geq d_i - |\mathcal{D}_1(\boldsymbol{u}_i)|, \text{ for } i \in [1, h]. \tag{3.23}$$

For any $5 \cdot 2^h$-tuple $\boldsymbol{q} \in Q$, let

$$L'(\boldsymbol{q}) \triangleq \sum_{a=1}^{5} \sum_{\boldsymbol{\sigma}\in B^h} \sum_{i\in\mathcal{D}^{(q_{a,\boldsymbol{\sigma}})}_{a,\boldsymbol{\sigma}}} |r_i|. \tag{3.24}$$

For any nonempty subset $Q' \subseteq Q$, let $\underline{L}'[Q']$ denote the optimal value of the following IPP:

$\mathcal{P}(Q')$: Minimize $\{L'(\boldsymbol{q}) \mid \boldsymbol{q} \in Q'\}$.

i.e., $\underline{L}'[Q'] \triangleq \min_{\boldsymbol{q} \in Q'} L'(\boldsymbol{q})$. A $5 \cdot 2^h$-tuple $\boldsymbol{q} \in Q'$ is called a **Q'-optimum** if $L'(\boldsymbol{q}) = \underline{L}'[Q']$. For convenience, we write $\underline{L}'[\phi] \triangleq +\infty$.

Then the minimum given by (3.2) is equal to the optimal value $\underline{L}'[Q^*]$ of the IPP $\mathcal{P}(Q^*)$ if the received sequence $\boldsymbol{r}$ satisfies (2.11).

Theorem 3.1: If the received sequence $\boldsymbol{r}$ satisfies (2.11), then we have the following formula:

$$\underline{L}'[Q^*] = \underline{L}[Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) \cap V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)]. \tag{3.25}$$

Proof. One hand, for any $\boldsymbol{u} \in Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) \cap V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$, from (3.12), (3.15), (3.16) and (3.19), we see that the $5 \cdot 2^h$-tuple $\boldsymbol{q}(\boldsymbol{u}) \triangleq (q(\boldsymbol{u})_{a,\boldsymbol{\sigma}} : a \in [1,5]$ and $\boldsymbol{\sigma} \in B^h)$ belongs to Q^*. From (3.15) and (3.24), we see $L(\boldsymbol{u}) \geq L'(\boldsymbol{q}(\boldsymbol{u}))$ and thus

$$\underline{L}'[Q^*] \leq \underline{L}[Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) \cap V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)]. \tag{3.26}$$

On the other hand, for any $5 \cdot 2^h$-tuple $\boldsymbol{q} \in Q$, let $\boldsymbol{u}(\boldsymbol{q})$ denote the sequence in V^N which satisfies

$$\mathcal{D}_1(\boldsymbol{u}(\boldsymbol{q})) = \bigcup_{a=1}^{5} \bigcup_{\boldsymbol{\sigma} \in B^h} \mathcal{D}^{(q_{a,\boldsymbol{\sigma}})}_{a,\boldsymbol{\sigma}}. \tag{3.27}$$

Then, from (3.15) and (3.20), we see that

$$q_{a,\boldsymbol{\sigma}} = |\mathcal{D}^{(q_{a,\boldsymbol{\sigma}})}_{a,\boldsymbol{\sigma}}| = |\mathcal{D}_1(\boldsymbol{u}(\boldsymbol{q})) \cap \mathcal{D}_{a,\boldsymbol{\sigma}}| = q(\boldsymbol{u}(\boldsymbol{q}))_{a,\boldsymbol{\sigma}}, \tag{3.28}$$

for $a \in [1,5]$ and $\boldsymbol{\sigma} \in B^h$. Thus from (3.10), (3.11), (3.15), (3.16) and (3.18), we see that the sequence $\boldsymbol{u}(\boldsymbol{q})$ must belong to the set $Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) \cap V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$. From (3.24) and (3.27), we see that $L'(\boldsymbol{q}) = L(\boldsymbol{u}(\boldsymbol{q}))$. Hence we have

$$\underline{L}'[Q^*] \geq \underline{L}[Y \cap V^N_{d_1, d_2, \ldots, d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)], \tag{3.29}$$

and thus (3.25) follows. □

3.2 Solving the Integer Programming Problem

In this section, the IPP $\mathcal{P}(Q^*)$ is solved for three simplest cases: i) $h = 0$, ii) $h = 1$ and $l_2 = N$, iii) $h = 1$ and $l_1 = l_2$. For the general case, we only give Lemma 3.1 which may be useful in simplifying the IPP. The IPP of the particular case $Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) = V^N$ will be investigated in Chapter 4.

At first, we define $t_1' \triangleq t_1 - |I_3|$, $t_2' \triangleq t_2 - |I_5|$ and $\delta_1 \triangleq d_1 - |\mathcal{D}_1(\boldsymbol{u}_1)|$.

i) Solving the IPP $\mathcal{P}(Q^*)$ for the case $h = 0$:

For this case, the set Q^* consists of the 5-tuples $\boldsymbol{q} = (q_1, q_2, \ldots, q_5)$ which satisfy

$$\begin{cases} 0 \leq q_a \leq |I_a|, \text{ for } a \in [1,5], \\ q_2 - q_3 \geq t_1', \\ q_4 - q_5 = t_2', \end{cases} \tag{3.30}$$

and thus

$$\underline{L}'[Q^*] = \sum_{i \in I_2^{(t_1')} \cup I_4^{(t_2')} \cup I_5^{(-t_2')}} |r_i|. \tag{3.31}$$

□

For $h \geq 1$, it is not such an easy work to solve the IPP because the number of variables increases rapidly when h grows. There is no known algorithm which solves this kind of nonlinear IPP effectively. However, the difficulty for solving the IPP may be reduced by splitting it into some simpler ones with less variables. The simple properties shown in the following lemma may be useful for the splitting of the IPP.

Lemma 3.1: If $\boldsymbol{q}^* \in Q^*$ is a Q^*-optimum such that there is no other $\boldsymbol{q}$ in Q^* such that

$$q_{a,\boldsymbol{\sigma}} \leq q^*_{a,\boldsymbol{\sigma}}, \text{ for } a = 1, 2, \ldots, 5 \text{ and all } \boldsymbol{\sigma} \in B^h, \tag{3.32}$$

then for any two tuples $\boldsymbol{\sigma}'$ and $\boldsymbol{\sigma}''$ with

$$\sigma_i' + \sigma_i'' \geq 1, \text{ for } i = 1, 2, \ldots, h, \tag{3.33}$$

we have

$$q^*_{i',\boldsymbol{\sigma}'} \cdot q^*_{i'',\boldsymbol{\sigma}''} = 0, \text{ for } (i', i'') \in \{(1,1), (3,3), (2,3), (4,5)\}. \tag{3.34}$$

Proof: We prove (3.34) only for $(i', i'') = (4, 5)$, the remaining cases can be proved similarly.

If $q^*_{4,\sigma'} \cdot q^*_{5,\sigma''} \neq 0$ for two tuples σ' and σ'' with (3.33), then the $5 \cdot 2^h$-tuple q with

$$q_{4,\sigma'} = q^*_{4,\sigma'} - 1, \quad q_{5,\sigma''} = q^*_{5,\sigma''} - 1, \tag{3.35}$$

$$q_{a,\sigma} = q^*_{a,\sigma}, \text{ for all } \sigma \in B^h \setminus \{\sigma', \sigma''\} \tag{3.36}$$

is also in the set Q^*. From $L'(q) \leq L'(q^*)$, we see that q is a Q^*-optimum too, contradicts our assumption. (3.34) with $(i', i'') = (4, 5)$ has been proved. □

ii) Solving the IPP $\mathcal{P}(Q^*)$ for the case $h = 1$ and $l_2 = N$:

For this case, the set Q^* consists of the 6-tuples $q = (q_{1,0}, q_{1,1}, q_{2,0}, q_{2,1}, q_{3,0}, q_{3,1})$ which satisfy

$$\begin{cases} 0 \leq q_{a,0} \leq |\mathcal{D}_{a,0}| \text{ and } 0 \leq q_{a,1} \leq |\mathcal{D}_{a,1}|, \text{ for } a = 1, 2, 3, \\ q_{2,0} + q_{2,1} - q_{3,0} - q_{3,1} \geq t'_1, \\ \sum_{a=1}^{3} (q_{a,0} - q_{a,1}) \geq \delta_1. \end{cases} \tag{3.37}$$

If $t'_1 \leq 0$ and $\delta_1 \leq 0$, then we see that $\underline{L}'[Q^*] = 0$, since $\mathbf{0}$ is in Q^* for this case. Below we assume that either $t'_1 > 0$ or $\delta_1 > 0$ holds. From Lemma 3.1, it is sufficient only to consider the tuples q in Q^* which satisfy $q_{1,1} = 0$. For $t \geq 0$, let $Q(t)$ denote the tuples in Q^* which satisfy $q_{1,1} = 0$ and $q_{1,0} = t$. Similar to the equality (4.41), we can get that

$$\underline{L}'[Q(t)] = \sum_{i \in \mathcal{T}(t) \cup \mathcal{D}_{1,0}^{(t)}} |r_i|, \tag{3.38}$$

where $\mathcal{T}(t)$ is defined as

$$\mathcal{T}(t) \triangleq \begin{cases} (\mathcal{D}_{2,0} \cup \mathcal{D}_{3,0}^{(\lfloor (\delta_1 - t - t'_1)/2 \rfloor)})^{(\delta_1 - t)}, & \text{if } t \leq \delta_1 - t'_1, \\ (\mathcal{D}_{2,0} \cup \mathcal{D}_{2,1}^{(\lfloor (t'_1 + t - \delta_1)/2 \rfloor)})^{(t'_1)}, & \text{otherwise.} \end{cases} \tag{3.39}$$

Case 1. $\delta_1 \geq t'_1$.

Since $|r_i| \leq |r_j|$ for any $i \in \mathcal{D}_{1,0}$ and $j \in \mathcal{D}_{2,0} \cup \mathcal{D}_{3,0}$, we have

$$\min_{0 \leq t \leq \delta_1 - t'_1} \underline{L}'[Q(t)] = \underline{L}'[Q(\min\{|\mathcal{D}_{1,0}|, \delta_1, \delta_1 - t'_1\})]. \tag{3.40}$$

Sine $\mathcal{T}(t) = \mathcal{T}(\delta_1 + t_1')$ for $t \geq \delta_1 + t_1'$ and $\mathcal{T}(\delta_1 - t_1' + 2t') = \mathcal{T}(\delta_1 - t_1' + 2t' + 1)$ for $0 \leq t' \leq t_1'$, there is an integer t_0 with $0 \leq t_0 \leq t_1'$ such that

$$\min_{t \geq \delta_1 - t_1'} \underline{L}'[Q(t)] = \underline{L}'[Q(\delta_1 - t_1' + 2t_0)]. \tag{3.41}$$

Hence,

$$\underline{L}'[Q^*] = \begin{cases} \underline{L}'[Q(\min\{|\mathcal{D}_{1,0}|, \delta_1\})], & \text{if } \min\{|\mathcal{D}_{1,0}|, \delta_1\} \leq \delta_1 - t_1', \\ \underline{L}'[Q(\delta_1 - t_1' + 2t_0)], & \text{otherwise.} \end{cases} \tag{3.42}$$

For the case of $\min\{|\mathcal{D}_{1,0}|, \delta_1\} > \delta_1 - t_1'$, it is needed to determine the integer t_0. For this case, $t_1' > 0$. Since $\underline{L}'[Q(\delta_1 - t_1' + 2(t' + 1))] - \underline{L}'[Q(\delta_1 - t_1' + 2t')]$ is monotonously increasing for $t' \geq 0$, the integer t_0 can be determined by at most $4t_1'$ operations of additions and comparisons of real numbers.

Case 2. $\delta_1 \leq t_1'$.

Similar to Case 1, there is an integer t_0 with $\lceil (t_1' - \delta_1)/2 \rceil \leq t_0 \leq t_1'$ such that

$$\underline{L}'[Q^*] = \underline{L}'[Q(\delta_1 - t_1' + 2t_0)]. \tag{3.43}$$

The integer t_0 can also be determined by at most $4(t_1' - \lceil (t_1' - \delta_1)/2 \rceil) = 4\lfloor (t_1' + \delta_1)/2 \rfloor$ operations of additions and comparisons of real numbers. □

iii) Solving the IPP $\mathcal{P}(Q^*)$ for the case $h = 1$ and $l_1 = l_2$:

For this case, the set Q^* consists of the 6-tuples $\boldsymbol{q} = (q_{1,0}, q_{1,1}, q_{4,0}, q_{4,1}, q_{5,0}, q_{5,1})$ which satisfy

$$\begin{cases} 0 \leq q_{a,0} \leq |\mathcal{D}_{a,0}| \text{ and } 0 \leq q_{a,1} \leq |\mathcal{D}_{a,1}|, \text{ for } a = 1, 4, 5, \\ q_{4,0} + q_{4,1} - q_{5,0} - q_{5,1} = t_2', \\ q_{1,0} + q_{4,0} + q_{5,0} - q_{1,1} - q_{4,1} - q_{5,1} \geq \delta_1. \end{cases} \tag{3.44}$$

From Lemma 3.1, it is sufficient only to consider the tuples $\boldsymbol{q}$ in Q^* which satisfy $q_{1,1} = 0$. For $t \geq 0$, let $Q'(t)$ denote the tuples in Q^* which satisfy $q_{1,1} = 0$ and $q_{1,0} = t$. Then, similar to the equality (4.120), we can show that

$$\underline{L}'[Q'(t)] = \sum_{i \in \mathcal{T}'(t) \cup \mathcal{D}_{1,0}^{(t)}} |r_i|, \tag{3.45}$$

where $\mathcal{T}'(t)$ is defined as

$$\mathcal{T}'(t) = \begin{cases} (\mathcal{D}_{5,0} \cup \mathcal{D}_{5,1}^{(\lfloor (t-t_2'-\delta_1)/2 \rfloor)})^{(-t_2')}, & \text{if } t_2' < 0 \text{ and } t \geq t_2' + \delta_1, \\ (\mathcal{D}_{4,0} \cup \mathcal{D}_{4,1}^{(\lfloor (t+t_2'-\delta_1)/2 \rfloor)})^{(t_2')}, & \text{if } t_2' \geq 0 \text{ and } t \geq \delta_1 - t_2', \\ \mathcal{D}_{4,0}^{(\lceil (t_2'+\delta_1-t)/2 \rceil)} \cup \mathcal{D}_{5,0}^{(\lceil (\delta_1-t_2'-t)/2 \rceil)}, & \text{otherwise.} \end{cases} \tag{3.46}$$

Let

$$t^{\mathrm{m}} \triangleq \max\{\delta_1 + t_2', \delta_1 - t_2'\}, \tag{3.47}$$

$$t_{\mathrm{m}} \triangleq \min\{\delta_1 + t_2', \delta_1 - t_2'\}, \tag{3.48}$$

$$t^* \triangleq \min\{\delta_1, \delta_1 - t_2', \lceil (\delta_1 - t_2' + |\mathcal{D}_{1,0}|)/2 \rceil\}, \tag{3.49}$$

$$\bar{t} \triangleq \min\{\lfloor (\delta_1 - t_2')/2 \rfloor, |\mathcal{D}_{5,0}|, |\mathcal{D}_{4,0}| - t_2'\}. \tag{3.50}$$

Then we have

$$\underline{L}'[Q'(t)] \geq \underline{L}'[Q'(t^{\mathrm{m}})], \quad \text{for } t \geq t^{\mathrm{m}}, \tag{3.51}$$

$$\min_{0 \leq t \leq t_{\mathrm{m}}} \underline{L}'[Q'(t)] = \begin{cases} \underline{L}'[Q'(\delta_1 - t_2' - 2t^*)], & \text{if } t^* \leq \bar{t}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{3.52}$$

Since $\mathcal{T}'(t_{\mathrm{m}} + 2t') = \mathcal{T}'(t_{\mathrm{m}} + 2t' + 1)$ for $0 \leq t' \leq |t_2'|$, there is an integer t_0 with $0 \leq t_0 \leq |t_2'|$ such that

$$\min_{t_{\mathrm{m}} \leq t \leq t^{\mathrm{m}}} \underline{L}'[Q'(t)] = \underline{L}'[Q'(t_{\mathrm{m}} + 2t_0)]. \tag{3.53}$$

Thus, from (3.51) to (3.53),

$$\underline{L}'[Q^*] = \begin{cases} \min\{\underline{L}'[Q'(t_{\mathrm{m}} + 2t_0)], \underline{L}'[Q'(\delta_1 - t_2' - 2t^*)]\}, & \text{if } t^* \leq \bar{t}, \\ \underline{L}'[Q'(t_{\mathrm{m}} + 2t_0)], & \text{otherwise.} \end{cases} \tag{3.54}$$

Since $\underline{L}'[Q(t_{\mathrm{m}} + 2(t' + 1))] - \underline{L}'[Q(t_{\mathrm{m}} + 2t')]$ is monotonously increasing for $0 \leq t' < |t_2'|$, the integer t_0 can be determined by at most $4 \cdot (t^{\mathrm{m}} - t_{\mathrm{m}})/2 = 4|t_2'|$ operations of additions and comparisons. □

Chapter 4

Integer Programming Problem Related to the Optimality Condition

4.1 The Integer Programming Problem

As pointed out in Section 2.2, the minimum $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)]$ can give an optimality condition or determine a region where the optimal codeword must fall in and it can be incorporated in any IDA which are based on the generation of a series codeword candidates. In this chapter, we consider to evaluate $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)]$.

Since $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)]$ is a particular case of (3.1) with $t_1 = 0$ and $l_2 = N$, it may be expressed as the optimal value of an IPP shown in Section 3.1. Because of $Y(\boldsymbol{v}, t_1, l_1, t_2, l_2) = V^N$, the IPP can be simplified to one with only 2^h variables. We first consider to deduce the simplified IPP.

Let Q_h denote the set of all 2^h-tuples over nonnegative integers. The 2^h components of $\boldsymbol{q} \in Q_h$ are referred as $q_{\boldsymbol{\sigma}}$ with $\boldsymbol{\sigma} \in B^h$. For $\boldsymbol{\sigma} \in B^h$, define

$$\mathcal{D}_{\boldsymbol{\sigma}} \triangleq \bigcap_{i=1}^{h} \mathcal{D}_{\sigma_i}(\boldsymbol{u}_i),\ n_{\boldsymbol{\sigma}} \triangleq |\mathcal{D}_{\boldsymbol{\sigma}}|. \tag{4.1}$$

Let Q_h^* denote the set of all 2^h-tuples $\boldsymbol{q} \in Q_h$ which satisfy

$$0 \le q_{\boldsymbol{\sigma}} \le n_{\boldsymbol{\sigma}},\ \text{for } \boldsymbol{\sigma} \in B^h, \tag{4.2}$$

$$\sum_{\sigma \in B^h} q_{\sigma}(-1)^{\sigma_i} \geq \delta_i, \text{ for } i = 1, 2, \ldots, h, \tag{4.3}$$

where

$$\delta_i \triangleq d_i - |\mathcal{D}_1(\boldsymbol{u}_i)|, \text{ for } i = 1, 2, \ldots, h. \tag{4.4}$$

For $\boldsymbol{q} \in Q_h$, let

$$\mathcal{D}(\boldsymbol{q}) \triangleq \bigcup_{\sigma \in B^h} \mathcal{D}_{\sigma}^{(q_{\sigma})}, \tag{4.5}$$

$$L'(\boldsymbol{q}) \triangleq \sum_{i \in \mathcal{D}(\boldsymbol{q})} |r_i|. \tag{4.6}$$

Then the minimum $\underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)]$ is equal to the optimal value of the following IPP:

$$\mathcal{P}(Q_h^*): \text{ Minimize } \{L'(\boldsymbol{q}) | \boldsymbol{q} \in Q_h^*\}.$$

We have the following theorem.

Theorem 4.1: If the received sequence $\boldsymbol{r}$ satisfies (2.11), then we have

$$\underline{L}'[Q_h^*] = \underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)]. \tag{4.7}$$

Proof: This theorem can be proved by simply following the proof of Theorem 3.1. The concrete proof for this theorem is omitted here. □

If $\delta_j \leq 0$ for $j = 1, 2, \ldots, h$, then $\underline{L}'[Q_h^*] = 0$ and the hard-decision tuple $\boldsymbol{z}$ is in the set $V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$. Below we assume that $\max_{1 \leq j \leq h} \delta_j > 0$. For convenience, we assume further, without loss of generality, that

$$\delta_1 \geq \delta_2 \geq \cdots \geq \delta_h \text{ and } \delta_1 > 0. \tag{4.8}$$

The IPP $\mathcal{P}(Q_h^*)$ has been investigated for the cases $h \leq 3$ in [14]-[16]. Here we mainly consider to solve the IPP $\mathcal{P}(Q_4^*)$. In Section 4.2, we split the IPP $\mathcal{P}(Q_h^*)$ into some simpler sub-IPPs for $1 \leq h \leq 5$ by classifying the minimal tuples of Q_h^*. In Sections 4.3 and 4.4, the IPP $\mathcal{P}(Q_h^*)$ is solved for the cases $h \leq 3$ with new methods though the main results have almost been obtained in [14]-[16]. In Section 4.5, an effective algorithm for solving the IPP $\mathcal{P}(Q_4^*)$ is presented. The number of additions and comparisons of real numbers is shown to be of order N for $h \leq 3$, and of order N^2 for $h = 4$. In Section 4.6, an improvement of Lemma 2.1 is presented.

4.2 Splitting the Integer Programming Problem

Since the object function of the IPP $\mathcal{P}(Q_h^*)$ is nonlinear, there is no known method which solves this kind of IPP effectively for large h. However, the object function $L'(\boldsymbol{q})$ possesses some special properties which can be used to design some algorithms for solving the IPP $\mathcal{P}(Q_h^*)$ with acceptable computational complexity. For example, $L'(\boldsymbol{q})$ is nondecreasing and convex on each variable (component of $\boldsymbol{q}$), and

$$L'(\boldsymbol{q}+\boldsymbol{q}') = L'(\boldsymbol{q}) + L'(\boldsymbol{q}') \tag{4.9}$$

for any two 2^h-tuples $\boldsymbol{q}, \boldsymbol{q}' \in Q_h^*$ with

$$q_{\boldsymbol{\sigma}} \cdot q'_{\boldsymbol{\sigma}} = 0 \quad \text{for all } \boldsymbol{\sigma} \in B^h. \tag{4.10}$$

These properties may be used to split the IPP $\mathcal{P}(Q_h^*)$ into some simpler sub-IPPs, each sub-IPP has less variables and thus may be solved more easily as compared with the IPP $\mathcal{P}(Q_h^*)$.

For tuples $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$ over integers with the same length, we write $\boldsymbol{\gamma} \geq \boldsymbol{\gamma}'$ if $\gamma_j \geq \gamma'_j$ for all indices j. We also denote $\boldsymbol{\gamma} > \boldsymbol{\gamma}'$ if $\boldsymbol{\gamma} \geq \boldsymbol{\gamma}'$ and $\boldsymbol{\gamma} \neq \boldsymbol{\gamma}'$. For $\boldsymbol{q}$ and $\boldsymbol{q}'$ in Q_h with $\boldsymbol{q} \geq \boldsymbol{q}'$, we see that $L'(\boldsymbol{q}) \geq L'(\boldsymbol{q}')$ from (2.11). Assume Q' is a subset of Q_h. A 2^h-tuple $\boldsymbol{q}' \in Q'$ is called an **Q'-optimum** if $L'(\boldsymbol{q}') = \underline{L}'[Q']$. A 2^h-tuple $\boldsymbol{q}' \in Q'$ is said to be **minimal** in Q' if $\boldsymbol{q}' \not> \boldsymbol{q}$ for all $\boldsymbol{q} \in Q'$. For any $\boldsymbol{q} \in Q_h$, let $S(\boldsymbol{q})$ denote the support set of $\boldsymbol{q}$, i.e.

$$S(\boldsymbol{q}) \triangleq \{\boldsymbol{\sigma} \in B^h : q_{\boldsymbol{\sigma}} \neq 0\}. \tag{4.11}$$

Clearly, the size of the support set of a minimal tuple in Q_h^* must be less than 2^h. The union of the support sets $S(\boldsymbol{q})$ with $\boldsymbol{q} \in Q'$ is called the support set of Q'.

Let $Q_h^{\min}$ denote the set of minimal tuples of Q_h^*. Then, we see easily that

$$\underline{L}'[Q_h^*] = \underline{L}'[Q_h^{\min}]. \tag{4.12}$$

For $h \leq 5$, we will show that the set $Q_h^{\min}$ can be classified into a few subsets, the size of the support set of each subset is only 2^{h-1}. Thus, to solve the IPP $\mathcal{P}(Q_h^*)$, it is sufficient only to solve some sub-IPPs, each sub-IPP has only 2^{h-1} variables. Below we will determine the set $Q_h^{\min}$ for $h \leq 5$.

Let Λ_0^h (or Λ_1^h) denote the set of h-tuples $\boldsymbol{\gamma}$ over even (or odd) integers with $\boldsymbol{\gamma} \geq \boldsymbol{\delta}$. Clearly, for all $\boldsymbol{q} \in Q_h^*$, we have

$$\sum_{\boldsymbol{\sigma} \in B^h} q_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \in \Lambda_0^h \cup \Lambda_1^h, \tag{4.13}$$

$$\boldsymbol{\gamma}(\boldsymbol{q}) \triangleq \sum_{\boldsymbol{\sigma} \in B^h} q_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} - \boldsymbol{\delta} \geq \mathbf{0}, \tag{4.14}$$

where $(-1)^{\boldsymbol{\sigma}} \triangleq ((-1)^{\sigma_1}, (-1)^{\sigma_2}, \ldots, (-1)^{\sigma_h})$ and $\boldsymbol{\delta} \triangleq (\delta_1, \delta_2, \ldots, \delta_h)$. Let Γ_0^h denote the set of h-tuples $\boldsymbol{\gamma}$ over nonnegative integers for which $\gamma_j = 0$ holds for at least one index j.

Lemma 4.1: Assume that $\boldsymbol{q}$ is a tuple in $Q_h^{\min}$. For any $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in B^h$,

$$\boldsymbol{\sigma}' \notin S(\boldsymbol{q}), \text{ if } (-1)^{\boldsymbol{\sigma}'} \leq \boldsymbol{\gamma}(\boldsymbol{q}), \tag{4.15}$$

$$\{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\} \not\subseteq S(\boldsymbol{q}), \text{ if } (-1)^{\boldsymbol{\sigma}'} + (-1)^{\boldsymbol{\sigma}''} \leq \boldsymbol{\gamma}(\boldsymbol{q}), \tag{4.16}$$

$$\{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\} \not\subseteq S(\boldsymbol{q}), \text{ if } \boldsymbol{\sigma}' + \boldsymbol{\sigma}'' \geq \mathbf{1}. \tag{4.17}$$

Proof. If $\boldsymbol{\sigma}' \in S(\boldsymbol{q})$ satisfies $(-1)^{\boldsymbol{\sigma}'} \leq \boldsymbol{\gamma}(\boldsymbol{q})$, then the tuple $\boldsymbol{q}'$ with

$$q'_{\boldsymbol{\sigma}'} = q_{\boldsymbol{\sigma}'} - 1, q'_{\boldsymbol{\sigma}} = q_{\boldsymbol{\sigma}} \text{ for all } \boldsymbol{\sigma} \in B^4 \setminus \{\boldsymbol{\sigma}'\},$$

is also in Q_h^*, which contradicts the assumption $\boldsymbol{q} \in Q_h^{\min}$.

If there are two tuples $\boldsymbol{\sigma}', \boldsymbol{\sigma}''$ in $S(\boldsymbol{q})$ such that $(-1)^{\boldsymbol{\sigma}'} + (-1)^{\boldsymbol{\sigma}''} \leq \boldsymbol{\gamma}(\boldsymbol{q})$ holds, then the tuple $\boldsymbol{q}'$ with

$$q'_{\boldsymbol{\sigma}'} = q_{\boldsymbol{\sigma}'} - 1, \; q'_{\boldsymbol{\sigma}''} = q_{\boldsymbol{\sigma}''} - 1,$$
$$q'_{\boldsymbol{\sigma}} = q_{\boldsymbol{\sigma}}, \text{ for all } \boldsymbol{\sigma} \in B^4 \setminus \{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\},$$

is also in Q_h^*, which contradicts the assumption $\boldsymbol{q} \in Q_h^{\min}$.

If $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in B^h$ satisfy $\boldsymbol{\sigma}' + \boldsymbol{\sigma}'' \geq \mathbf{1}$, then we have

$$(-1)^{\boldsymbol{\sigma}'} + (-1)^{\boldsymbol{\sigma}''} \leq (-1)^{\mathbf{1}} \leq \mathbf{0} \leq \boldsymbol{\gamma}(\boldsymbol{q})$$

and from (4.16), we get $\{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\} \not\subseteq S(\boldsymbol{q})$. □

From (4.15), we have the following simple corollary.

Corollary 4.1: Assume that $\boldsymbol{q}$ is a tuple in $Q_h^{\min}$.
(1) The all-one h-tuple $\mathbf{1} \triangleq 11\cdots1$ is not in $S(\boldsymbol{q})$, or in other words, $w_{\mathrm{H}}(\boldsymbol{\sigma}) < h$ for all $\boldsymbol{\sigma} \in S(\boldsymbol{q})$, where $w_{\mathrm{H}}(\boldsymbol{\sigma})$ denotes the Hamming weight of $\boldsymbol{\sigma}$.
(2) There is at least one index j such that $\gamma(\boldsymbol{q})_j = 0$, i.e. $\boldsymbol{\gamma}(\boldsymbol{q}) \in \Gamma_0^h$.

□

The following lemma can be viewed as an generalization of (4.16).

Lemma 4.2: For $\boldsymbol{q} \in Q_h^*$, $\boldsymbol{q}$ belongs to $Q_h^{\min}$ if and only if

$$\sum_{\boldsymbol{\sigma}\in B^h} q'_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \not\leq \boldsymbol{\gamma}(\boldsymbol{q}), \text{ for any } \mathbf{0} < \boldsymbol{q}' \leq \boldsymbol{q}. \tag{4.18}$$

Proof: From the definition, we see that $\boldsymbol{q}$ belongs to $Q_h^{\min}$ if and only if

$$\sum_{\boldsymbol{\sigma}\in B^h} q''_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \not\geq \boldsymbol{\delta}, \text{ for any } \mathbf{0} \leq \boldsymbol{q}'' < \boldsymbol{q}. \tag{4.19}$$

Clearly, (4.19) is equivalent to

$$\sum_{\boldsymbol{\sigma}\in B^h} (q_{\boldsymbol{\sigma}} - q''_{\boldsymbol{\sigma}})(-1)^{\boldsymbol{\sigma}} \not\leq \sum_{\boldsymbol{\sigma}\in B^h} q_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} - \boldsymbol{\delta}, \text{ for any } \mathbf{0} \leq \boldsymbol{q}'' < \boldsymbol{q}.$$

This is equivalent to (4.18). □

For any $\boldsymbol{\gamma} \geq \mathbf{0}$, we call a subset $\Xi \subseteq B^h$ an **$m(\boldsymbol{\gamma})$-set** if

$$\sum_{\boldsymbol{\sigma}\in \Xi} q_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \not\leq \boldsymbol{\gamma}, \text{ for any } \boldsymbol{q} \in Q_h \text{ with } \sum_{\boldsymbol{\sigma}\in \Xi} q_{\boldsymbol{\sigma}} \geq 1. \tag{4.20}$$

For any $\boldsymbol{\sigma} \in B^h$, let $\overline{\boldsymbol{\sigma}}$ denote the tuple in B^h whose j-th component is $1-\sigma_j$ for all $j \in [1, h]$. The following lemma can be obtained easily from the definitions.

Lemma 4.3: (1) Any $m(\boldsymbol{\gamma})$-set Ξ is also an $m(\mathbf{0})$-set.
(2) If Ξ is an $m(\boldsymbol{\gamma})$-set, then $\mathbf{1} \notin \Xi$ and $\{\boldsymbol{\sigma}, \overline{\boldsymbol{\sigma}}\} \not\subseteq \Xi$.

□

From (2) of Lemma 4.3, we see that $|\Xi| \leq 2^{h-1}$ for all $m(\boldsymbol{\gamma})$-set Ξ. If $m(\boldsymbol{\gamma})$-set Ξ satisfies $|\Xi| = 2^{h-1}$, then we call Ξ an **$M(\boldsymbol{\gamma})$-set**.

Lemma 4.4: 1. Any $m(\mathbf{0})$-set is a subset of an $M(\mathbf{0})$-set.
2. For $1 \leq h \leq 5$ and $\boldsymbol{\gamma} \in \Gamma_0^h$, if Ξ is an $m(\boldsymbol{\gamma})$-set, then Ξ must be a subset of some $M(\boldsymbol{\gamma})$-set.

Proof: See Appendix B. □

For $1 \leq h \leq 5$, let $\aleph_h$ denote the set of $M(\mathbf{0})$-sets.

Lemma 4.5: For $1 \leq h \leq 5$ and $\Xi \in \aleph_h$, there is just one tuple, denoted $\boldsymbol{\sigma}(\Xi)$, in Ξ such that

$$\boldsymbol{\rho}(\Xi) \triangleq \boldsymbol{\delta} + \boldsymbol{\sigma}(\Xi) \in \Lambda_0^h \cup \Lambda_1^h. \tag{4.21}$$

Proof: Let $\mathcal{S}$ denote the set of the tuples $\boldsymbol{\sigma} \in B^h$ which satisfy

$$\boldsymbol{\delta} + \boldsymbol{\sigma} \in \Lambda_0^h \cup \Lambda_1^h.$$

It is clear that $\overline{\boldsymbol{\sigma}} \in \mathcal{S}$ if $\boldsymbol{\sigma} \in \mathcal{S}$. Let $\boldsymbol{\sigma}'$ and $\boldsymbol{\sigma}''$ be two tuples in $\mathcal{S}$. Clearly, $\boldsymbol{\sigma}' + \boldsymbol{\sigma}''$ is a tuple over even integers or a tuple over odd integers, and $\sigma_i' + \sigma_i'' \in \{0, 1, 2\}$ for all $i \in [1, h]$. If $\sigma_j' + \sigma_j'' \in \{0, 2\}$ for some index j, then for all $i \in [1, h]$ we have $\sigma_i' + \sigma_i'' \in \{0, 2\}$ and thus $\sigma_i' = \sigma_i''$, i.e. $\boldsymbol{\sigma}' = \boldsymbol{\sigma}''$. The remaining case is that $\sigma_i' + \sigma_i'' = 1$ holds for all $i \in [1, h]$. This means that $\boldsymbol{\sigma}'' = \overline{\boldsymbol{\sigma}'}$. Hence we have proved that $\mathcal{S}$ is a set of form $\{\boldsymbol{\sigma}', \overline{\boldsymbol{\sigma}'}\}$ with $\boldsymbol{\sigma}' \in B^h$. From (2) of Lemma 4.3, Ξ contains just one of the two tuples in $\mathcal{S}$. □

For $I \subseteq [1, h]$ and $\Xi \subseteq B^h$, let

$$p_I[\Xi] \triangleq \{p_I(\boldsymbol{\sigma}) : \boldsymbol{\sigma} \in \Xi\}. \tag{4.22}$$

For $\Xi \in \aleph_h$, let $I(\Xi)$ denote the smallest subset $I \subseteq [1, h]$ which satisfies

$$\Xi = \{\boldsymbol{\sigma} \in B^h : p_I(\boldsymbol{\sigma}) \in p_I[\Xi]\}. \tag{4.23}$$

Now we can prove the following theorem.

Theorem 4.2: For $1 \leq h \leq 5$,

$$Q_h^{\min} = \bigcup_{\Xi \in \aleph_h} Q_h(\Xi), \tag{4.24}$$

where $Q_h(\Xi)$ is the set of tuples $\boldsymbol{q} \in Q_h$ with $S(\boldsymbol{q}) \subseteq \Xi$ which satisfy (4.2) and

$$\begin{cases} \sum_{\boldsymbol{\sigma} \in \Xi} q_{\boldsymbol{\sigma}} (-1)^{\sigma_i} = \rho(\Xi)_i, & \text{for } i \in I(\Xi), \\ \sum_{\boldsymbol{\sigma} \in \Xi} q_{\boldsymbol{\sigma}} (-1)^{\sigma_i} \geq \rho(\Xi)_i, & \text{for } i \in [1, h] \setminus I(\Xi). \end{cases} \tag{4.25}$$

Proof: See Appendix C. □

To solve the IPP $\mathcal{P}(Q_h)$, from (4.12) and (4.24), it is sufficient to solve the sub-IPPs $\mathcal{P}(Q_h(\Xi))$ for all $M(\mathbf{0})$-sets $\Xi \in \aleph_h$.

For positive integer h, let

$$\Theta_h \triangleq \{\boldsymbol{\sigma} \in B^h : w_{\mathrm{H}}(\boldsymbol{\sigma}) \leq 1\}. \tag{4.26}$$

The following theorem gives all of the $M(\mathbf{0})$-sets for the cases of $h \leq 5$.

Theorem 4.3: If $h = 1$, $\{0\}$ is the unique $M(\mathbf{0})$-set.
If $h = 2$, there are two $M(\mathbf{0})$-sets: $\{00, 01\}$ and $\{00, 10\}$.
If $h = 3$, there are four $M(\mathbf{0})$-sets: Θ_3 and

$$A_i \triangleq \{\boldsymbol{\sigma} \in B^3 : \sigma_i = 0\}, \text{ for } i = 1, 2, 3. \tag{4.27}$$

If $h = 4$, there are 12 $M(\mathbf{0})$-sets:

$$C_i \triangleq \{\boldsymbol{\sigma} \in B^4 : \sigma_i = 0\}, \text{ for } i = 1, 2, 3, 4, \tag{4.28}$$

$$D_i \triangleq \Theta_4 \cup \{\boldsymbol{\sigma} \in B^4 : w_{\mathrm{H}}(\boldsymbol{\sigma}) = 2,\ \sigma_i = 1\}, \text{ for } i = 1, 2, 3, 4, \tag{4.29}$$

$$E_i \triangleq \Theta_4 \cup \{\boldsymbol{\sigma} \in B^4 : w_{\mathrm{H}}(\boldsymbol{\sigma}) = 2,\ \sigma_i = 0\}, \text{ for } i = 1, 2, 3, 4. \tag{4.30}$$

If $h = 5$, let

$$\Theta^* \triangleq \{\boldsymbol{\sigma} \in B^5 : w_{\mathrm{H}}(\boldsymbol{\sigma}) = 2\}, \tag{4.31}$$

$$\mathcal{K} \triangleq \{(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in \Theta^* \times \Theta^* : \sigma'_i = \sigma''_i = 1 \text{ for at least one } i\}, \tag{4.32}$$

there are 81 $M(\mathbf{0})$-sets:

$$F \triangleq \Theta_5 \cup \Theta^*, \tag{4.33}$$

$$G_i \triangleq \{\boldsymbol{\sigma} \in B^5 : \sigma_i = 0\}, \text{ for } 1 \leq i \leq 5, \tag{4.34}$$

$$H_i \triangleq \Theta_5 \cup \{\boldsymbol{\sigma} \in B^5 : 2 \leq w_{\mathrm{H}}(\boldsymbol{\sigma}) \leq 3,\ \sigma_i = 0\}, \text{ for } 1 \leq i \leq 5, \tag{4.35}$$

$$I_{i_1 i_2 i_3} \triangleq \{\boldsymbol{\sigma} \in B^5 : \sum_{j=1}^{3} \sigma_{i_j} \leq 1\}, \text{ for } 1 \leq i_1 < i_2 < i_3 \leq 5, \tag{4.36}$$

$$J^{i_4}_{i_1 i_2 i_3} \triangleq \{\boldsymbol{\sigma} \in B^5 : \sum_{j=1}^{4} \sigma_{i_j} \leq 1\} \cup \{\boldsymbol{\sigma} \in B^5 : \sum_{j=1}^{3} \sigma_{i_j} = 2,\ \sigma_{i_4} = 0\},$$
$$\text{for } 1 \leq i_1 < i_2 < i_3 \leq 5 \text{ and } i_4 \in [1, 5] \setminus \{i_1, i_2, i_3\}, \tag{4.37}$$

$$K_{\boldsymbol{\sigma}', \boldsymbol{\sigma}''} \triangleq \Theta_5 \cup (\Theta^* \setminus \{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\}) \cup \{\overline{\boldsymbol{\sigma}'}, \overline{\boldsymbol{\sigma}''}\}, \text{ for } (\boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in \mathcal{K}. \tag{4.38}$$

Proof: See Appendix D. □

4.3 Solving the Integer Programming Problem for the Cases $h \leq 2$

At first, we assume that $h = 2$. Q_2^* consists of the tuples $\boldsymbol{q} \in Q_2$ which satisfy (4.2) and

$$\begin{cases} q_{00} + q_{01} - q_{10} - q_{11} \geq \delta_1, \\ q_{00} - q_{01} + q_{10} - q_{11} \geq \delta_2. \end{cases}$$

From (4.12), (4.24) and Theorem 4.3,

$$\underline{L}[Q_2^*] = \min\{\underline{L}'[Q_2(\{00,01\})], \underline{L}'[Q_2(\{00,10\})]\}. \tag{4.39}$$

For $\boldsymbol{q} \in Q_2(\{00,01\})$, we have

$$q_{00} + q_{01} = \delta_1 \text{ and } q_{00} - q_{01} \geq \delta_2,$$

and then

$$q_{00} + q_{01} = \delta_1 \text{ and } q_{01} \leq \lceil(\delta_1 - \delta_2)/2\rceil.$$

Thus we have

$$\underline{L}[Q_2(\{00,01\})] = \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1-\delta_2)/2\rfloor)})^{(\delta_1)}} |r_i|. \tag{4.40}$$

If $Q_2(\{00,10\}) \neq \phi$, let $\boldsymbol{q}$ be an arbitrary tuple in $Q_2(\{00,10\})$, then

$$q_{00} + q_{10} = \delta_2 \text{ and } q_{00} - q_{10} \geq \delta_1,$$

and thus by (4.8), we see that $q_{10} = 0$ and $\delta_2 = \delta_1$. Hence, $\boldsymbol{q} \in Q_2(\{00,01\})$ and then $Q_2(\{00,10\}) \subseteq Q_2(\{00,01\})$. Thus we have the following theorem.

Theorem 4.4: For the case $h = 2$,

$$\underline{L}[Q_2^*] = \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1-\delta_2)/2\rfloor)})^{(\delta_1)}} |r_i|. \tag{4.41}$$

□

According to (4.41), we see that the testing condition of the case of $h = 2$ can be computed with $\delta_1 - 1 \le N$ additions of real numbers.

If we put $\boldsymbol{u}_2 = \boldsymbol{u}_1$ and $d_2 = d_1$, then $\mathcal{D}_{00} = \mathcal{D}_0(\boldsymbol{u}_1)$ in (4.41). Then we get the following corollary.

Corollary 4.2: For the case $h = 1$,

$$\underline{L}[Q_1^*] = \sum_{i \in \mathcal{D}_0(\boldsymbol{u}_1)^{(\delta_1)}} |r_i|. \tag{4.42}$$

□

If the reference word $\boldsymbol{u}_1$ is a codeword, the special case of (4.42) with $d_1 = d_{\min}$ (the minimum distance of the code) is the same formula given by Taipale and Pursley [25].

4.4 Solving the Integer Programming Problem for the Case $h = 3$

Assume $h = 3$. Q_3^* consists of the tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} - q_{100} + q_{010} + q_{001} + q_{011} - q_{101} - q_{110} - q_{111} \ge \delta_1, \\ q_{000} + q_{100} - q_{010} + q_{001} - q_{011} + q_{101} - q_{110} - q_{111} \ge \delta_2, \\ q_{000} + q_{100} + q_{010} - q_{001} - q_{011} - q_{101} + q_{110} - q_{111} \ge \delta_3. \end{cases}$$

From (4.12), (4.24) and Theorem 4.3,

$$\underline{L}[Q_3^*] = \min\{\underline{L}'[Q_3(\Theta_3)], \underline{L}'[Q_3(A_1)], \underline{L}'[Q_3(A_2)], \underline{L}'[Q_3(A_3)]\}. \tag{4.43}$$

Since we have (4.8), the last two terms in the minimum of the right hand of (4.43) can be ignored. Indeed, for any tuple $\boldsymbol{q} \in Q_3(A_2)$, we have

$$\begin{cases} q_{010} = q_{110} = q_{011} = q_{111} = 0, \\ q_{000} - q_{100} + q_{001} - q_{101} \ge \delta_1, \\ q_{000} + q_{100} + q_{001} + q_{101} = \delta_2, \\ q_{000} + q_{100} - q_{001} - q_{101} \ge \delta_3, \end{cases} \tag{4.44}$$

and thus from (4.8) we see that $q_{100} = q_{101} = 0$ and $\delta_2 = \delta_1$. Hence, $\boldsymbol{q}$ belongs to $Q_3(A_1)$ and thus we have $Q_3(A_2) \subseteq Q_3(A_1)$. Similarly, we can prove that $Q_3(A_3) \subseteq Q_3(A_1)$. Then we have the following theorem.

Theorem 4.5: For the case $h = 3$,

$$\underline{L}[Q_3^*] = \min\{\underline{L}'[Q_3(\Theta_3)], \underline{L}'[Q_3(A_1)]\}. \tag{4.45}$$

□

We will present a sub-algorithm-$Q_3(\Theta_3)$ for solving the IPP $\mathcal{P}(Q_3(\Theta_3))$ in Section 4.4.1 and a Sub-algorithm-$Q_3(A_1)$ for solving the IPP $\mathcal{P}(Q_3(A_1))$ in Section 4.4.2. The number of operations of additions and comparisons of real numbers for either of the two sub-algorithms is not more than $5\delta_1 - 1$. Hence the number of additions and comparisons of real numbers for solving the IPP $\mathcal{P}(Q_3^*)$ is not more than $10\delta_1 - 1$, and thus is of order N.

4.4.1 Sub-algorithm for Solving the IPP $\mathcal{P}(Q_3(\Theta_3))$

In this subsection, we show a method for solving the IPP $\mathcal{P}(Q_3(\Theta_3))$. Firstly, we consider to give the optimal solution for a sub-IPP of IPP $\mathcal{P}(Q_3(\Theta_3))$, which is over a set of tuples $\boldsymbol{q} \in Q_3(\Theta_3)$ with fixed q_{011}. Then we show that $\underline{L}'[Q_3(\Theta_3)]$ can be computed iteratively from the optimal solution of such a sub-IPP.

Clearly, a tuple $\boldsymbol{q} \in Q_3^*$ belongs to $Q_3(\Theta_3)$ if and only if $\boldsymbol{q}$ satisfies (4.2) and

$$\begin{cases} q_{100} = q_{110} = q_{101} = q_{111} = 0, \\ q_{000} + q_{010} + q_{001} + q_{011} = \delta_1, \\ q_{010} + q_{011} \le \lfloor (\delta_1 - \delta_2)/2 \rfloor, \\ q_{001} + q_{011} \le \lfloor (\delta_1 - \delta_3)/2 \rfloor. \end{cases} \tag{4.46}$$

Since $q_{011} \le n_{011}$ and $q_{\boldsymbol{\sigma}} \ge 0$ for all $\boldsymbol{\sigma} \in B^3$, the q_{011} in (4.46) must satisfy

$$0 \le q_{011} \le k_0 \triangleq \min\{n_{011}, \delta_1, \lfloor (\delta_1 - \delta_2)/2 \rfloor, \lfloor (\delta_1 - \delta_3)/2 \rfloor\}. \tag{4.47}$$

For any k with $0 \le k \le k_0$, write $\Re(k) \triangleq \{\boldsymbol{q} \in Q_3(\Theta_3) : q_{011} = k\}$ and

$$X(k) \triangleq (\mathcal{D}_{000} \cup \mathcal{D}_{010}^{(\lfloor (\delta_1-\delta_2)/2 \rfloor - k)} \cup \mathcal{D}_{001}^{(\lfloor (\delta_1-\delta_3)/2 \rfloor - k)})^{(\delta_1 - k)}. \tag{4.48}$$

Then we see that $\Re(k)$ is not empty if and only if ∞ is not in $X(k)$. From the definition of $X(k)$, we can determine whether ∞ belongs $X(k)$ or not, and find $X(k+1)$ from $X(k)$ easily. If $\Re(k) \ne \phi$, let $\boldsymbol{q}^k$ denote the tuple in $\Re(k)$ which satisfies $\mathcal{D}(\boldsymbol{q}^k) = X(k) \cup \mathcal{D}_{011}^{(k)}$. Then $\boldsymbol{q}^k$ must be an $\Re(k)$-optimum. Furthermore, we have the following lemma.

Lemma 4.6: 1. If $Q_3(\Theta_3) \neq \phi$, then there are two integers k_1^{l} and k_1^{r} with $0 \leq k_1^{\mathrm{l}} \leq k_1^{\mathrm{r}} \leq k_0$ such that ∞ is not in $X(k)$ if and only if k satisfies $k_1^{\mathrm{l}} \leq k \leq k_1^{\mathrm{r}}$, this implies

$$\underline{L}'[Q_3(\Theta_3)] = \min_{k_1^{\mathrm{l}} \leq k \leq k_1^{\mathrm{r}}} L'(\boldsymbol{q}^k). \tag{4.49}$$

2. For k with $k_1^{\mathrm{l}} \leq k < k_1^{\mathrm{r}}$, let $\mu(k) \triangleq \sum_{i \in X(k)} |r_i| - \sum_{j \in X(k+1)} |r_j|$, then $\mu(k)$ is non-increasing, and thus there is an integer k_1 with $k_1^{\mathrm{l}} \leq k_1 \leq k_1^{\mathrm{r}}$ such that $L'(\boldsymbol{q}^k)$ is non-increasing for $k_1^{\mathrm{l}} \leq k \leq k_1$ and is non-decreasing for $k_1 \leq k \leq k_1^{\mathrm{r}}$.

Proof: See Appendix E. □

According to Lemma 4.6, we can evaluate $\underline{L}'[Q_3(\Theta_3)]$ by the following sub-algorithm.

Sub-algorithm-$Q_3(\Theta_3)$

Input. N-tuple $\boldsymbol{r}$ with (2.11). 3-tuple $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3)$ satisfying (4.8). $\mathcal{D}_{\boldsymbol{\sigma}}$ and $n_{\boldsymbol{\sigma}}$ defined by (4.1) for all $\boldsymbol{\sigma} \in \{000, 010, 001, 011\}$.

Output. $\underline{L}'[Q_3(\Theta_3)]$.

Step 1. Compute k_0 by (4.47) and set $k = 0$.

Step 2. If $k > k_0$, then output $+\infty$ (i.e. $Q_3(\Theta_3) = \phi$) and END, otherwise, determine $X(k)$ by (4.48). If $\infty \in X(k)$, then set $k = k + 1$ and goto Step 2, otherwise, goto Step 3.

Step 3. If $k = k_0$, then goto Step 4, otherwise, determine $X(k+1)$ by (4.48). If $\infty \in X(k+1)$ or $L'(\boldsymbol{q}^{k+1}) - L'(\boldsymbol{q}^k) > 0$, then goto Step 4, otherwise, set $k = k+1$ and then goto Step 3.

Step 4. Output $L'(\boldsymbol{q}^k)$ and END. □

Consider the computational complexity of this sub-algorithm. To determine $X(k)$ and whether ∞ belongs $X(k)$ or not, no operation of real numbers is needed. To determine whether $L'(\boldsymbol{q}^{k+1}) - L'(\boldsymbol{q}^k) > 0$ is valid or not, it needs at most 4 additions and comparisons of real numbers. Since $k_0 \leq \delta_1$ and the number of additions and comparisons of real numbers for computing $L'(\boldsymbol{q}^k)$ is $\delta_1 - 1$, we see the number of additions and comparisons of real numbers for the evaluating of $\underline{L}'[Q_3(\Theta_3)]$ by the Sub-algorithm-$Q_3(\Theta_3)$ is not more than $5\delta_1 - 1$.

4.4.2 Sub-algorithm for Solving the IPP $\mathcal{P}(Q_3(A_1))$

A method for solving the IPP $\mathcal{P}(Q_3(A_1))$ is shown in this subsection.

Clearly, a tuple $\boldsymbol{q} \in Q_3^*$ belongs to $Q_3(A_1)$ if and only if $\boldsymbol{q}$ satisfies (4.2) and

$$\begin{cases} q_{000} + q_{100} = \delta_1' \triangleq (\rho(A_1)_2 + \rho(A_1)_3)/2, \\ q_{000} + q_{010} = \delta_2' \triangleq (\rho(A_1)_1 + \rho(A_1)_3)/2, \\ q_{000} + q_{001} = \delta_3' \triangleq (\rho(A_1)_1 + \rho(A_1)_2)/2, \\ q_{011} = q_{101} = q_{110} = q_{111} = 0. \end{cases} \tag{4.50}$$

Write

$$k_2^{\mathrm{l}} \triangleq \max\{\delta_1' - n_{100}, \delta_2' - n_{010}, \delta_3' - n_{001}\}, \tag{4.51}$$

$$k_2^{\mathrm{r}} \triangleq \min\{n_{000}, \delta_1', \delta_2', \delta_3'\}. \tag{4.52}$$

We see easily that $Q_3(A_1) \neq \phi$ if and only if $k_2^{\mathrm{l}} \leq k_2^{\mathrm{r}}$. If $Q_3(A_1) \neq \phi$, i.e. $k_2^{\mathrm{l}} \leq k_2^{\mathrm{r}}$, then the set $Q_3(A_1)$ consists of the tuples $\dot{\boldsymbol{q}}^k$ with $k_2^{\mathrm{l}} \leq k \leq k_2^{\mathrm{r}}$ and

$$\begin{cases} \dot{q}_{100}^k = \delta_1' - k, \ \dot{q}_{010}^k = \delta_2' - k, \ \dot{q}_{001}^k = \delta_3' - k, \\ \dot{q}_{000}^k = k, \ \dot{q}_{011}^k = \dot{q}_{101}^k = \dot{q}_{110}^k = \dot{q}_{111}^k = 0. \end{cases} \tag{4.53}$$

Since $L'(\dot{\boldsymbol{q}}^{k+1}) - L'(\dot{\boldsymbol{q}}^k)$ is non-decreasing, the IPP $\mathcal{P}(Q_3(A_1))$ can be solved by the following Sub-algorithm-$Q_3(A_1)$.

Sub-algorithm-$Q_3(A_1)$
Input. N-tuple $\boldsymbol{r}$ with (2.11). 3-tuple $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3)$ satisfying (4.8). $\mathcal{D}_{\boldsymbol{\sigma}}$ and $n_{\boldsymbol{\sigma}}$ defined by (4.1) for all $\boldsymbol{\sigma} \in \{000, 100, 010, 001\}$.
Output. $\underline{L}'[Q_3(A_1)]$.
Step 1. Determine the sequence $\rho(A_1)$ and compute the integers δ_1', δ_2' and δ_3' by (4.50). Then, compute k_2^{l} and k_2^{r} by (4.51) and (4.52), respectively. If $k_2^{\mathrm{l}} > k_2^{\mathrm{r}}$, output $+\infty$ (i.e. $Q_3(A_1) = \phi$) and END, otherwise, set $k = k_2^{\mathrm{l}}$.
Step 2. If $k = k_2^{\mathrm{r}}$ or $L'(\dot{\boldsymbol{q}}^{k+1}) - L'(\dot{\boldsymbol{q}}^k) > 0$, goto Step 3, otherwise, set $k = k + 1$ and goto Step 2.
Step 3. Output $L'(\dot{\boldsymbol{q}}^k)$ and END. □

With 4 operations of additions and comparisons of real numbers we can determine whether $L'(\dot{\boldsymbol{q}}^{k+1}) - L'(\dot{\boldsymbol{q}}^k) > 0$ or not. Since $k_2^{\mathrm{r}} \leq \delta_1$ and the number of additions and comparisons of real numbers for computing $L'(\dot{\boldsymbol{q}}^k)$ is

$\delta_1' + \delta_2' + \delta_3' - 2k - 1 \leq 3\delta_1 - 2k - 1$, we see the number of additions and comparisons of real numbers for the solving of the IPP $\mathcal{P}(Q_3(A_1))$ by the Sub-algorithm-$Q_3(A_1)$ is not more than $5\delta_1 - 1$ too.

4.5 Solving the Integer Programming Problem for the Case $h = 4$

Assume $h = 4$. From Theorem 4.3, $\aleph_4 = \bigcup_{\ell=1}^{4}\{C_\ell, D_\ell, E_\ell\}$. Similar to the cases $h \leq 3$, for some $M(\mathbf{0})$-sets $\Xi \in \aleph_4$ the sets $Q_4(\Xi)$ may be empty or covered by the others and thus should be excluded from further consideration.

Lemma 4.7: 1. If $Q_4(C_\ell) \neq \phi$, then $\delta_\ell \geq \max_{j\neq\ell}\{\delta_j\}$.
2. If $Q_4(D_\ell) \neq \phi$, then $\max\{\rho(D_\ell)_\ell, \max_{j\neq\ell}\{|\rho(D_\ell)_j|\}\} \leq \sum_{j\neq\ell}\rho(D_\ell)_j$.
3. If $Q_4(E_\ell) \neq \phi$, then $\sum_{j=1}^{4}\rho(E_\ell)_j \geq 0$ and $\rho(E_\ell)_\ell + \min_{j\neq\ell}\{\rho(E_\ell)_j\} \geq 0$.

Proof. 1. Assume $\boldsymbol{q} \in Q_4(C_\ell)$. For each j with $j \neq \ell$, we have $\delta_\ell = \sum_{\boldsymbol{\sigma}\in C_\ell} q_{\boldsymbol{\sigma}} \geq \sum_{\boldsymbol{\sigma}\in C_\ell} q_{\boldsymbol{\sigma}}(-1)^{\sigma_j} \geq \delta_j$.

2. Assume $\boldsymbol{q} \in Q_4(D_\ell)$. Since $\sum_{j\neq\ell}(-1)^{\sigma_j} \geq 1$ for all $\boldsymbol{\sigma} \in D_\ell$, we see $\sum_{j\neq\ell}\rho(D_\ell)_j = \sum_{j\neq\ell}\sum_{\boldsymbol{\sigma}\in D_\ell} q_{\boldsymbol{\sigma}}(-1)^{\sigma_j} \geq \sum_{\boldsymbol{\sigma}\in D_\ell} q_{\boldsymbol{\sigma}} \geq \max_{1\leq j\leq 4}\{|\sum_{\boldsymbol{\sigma}\in D_\ell} q_{\boldsymbol{\sigma}} (-1)^{\sigma_j}|\} \geq \max\{\rho(D_\ell)_\ell, \max_{j\neq\ell}\{|\rho(D_\ell)_j|\}\}$.

3. Assume $\boldsymbol{q} \in Q_4(E_\ell)$. Since for all $\boldsymbol{\sigma} \in E_\ell$ we have $\sum_{j=1}^{4}(-1)^{\sigma_j} \geq 0$ and $(-1)^{\sigma_\ell} + \min_{j\neq\ell}\{(-1)^{\sigma_j}\} \geq 0$, we see $\sum_{j=1}^{4}\rho(E_\ell)_j = \sum_{j=1}^{4}\sum_{\boldsymbol{\sigma}\in E_\ell} q_{\boldsymbol{\sigma}}(-1)^{\sigma_j} \geq 0$ and $\rho(E_\ell)_\ell + \min_{j\neq\ell}\{\rho(E_\ell)_j\} = \sum_{\boldsymbol{\sigma}\in E_\ell} q_{\boldsymbol{\sigma}}(-1)^{\sigma_\ell} + \min_{j\neq\ell}\{\sum_{\boldsymbol{\sigma}\in E_\ell} q_{\boldsymbol{\sigma}}(-1)^{\sigma_j}\} \geq 0$. □

Theorem 4.6: If the 4-tuple $\boldsymbol{\delta}$ satisfies (4.8), then we have

$$Q_4^{\min} = \bigcup_{\Xi\in\{C_1\}\cup\aleph_4^*} Q_4(\Xi), \tag{4.54}$$

where $\aleph_4^*$ is a subset of $\aleph_4$ defined as follows:
(i). ϕ if $\delta_2 + \delta_3 + 1 < 0$ and $\sum_{j=1}^{4}\rho(E_1)_j < 0$;
(ii). $\{E_1\}$ if $\delta_2 + \delta_3 + 1 < 0$ and $\sum_{j=1}^{4}\rho(E_1)_j \geq 0$;
(iii). $\{D_4\}$ if $\delta_2 + \delta_3 + 1 \geq 0$ and $\delta_1 + \delta_4 + 1 < 0$;
(iv). $\{E_1, D_4\}$ if $\delta_2 + \delta_3 + 1 \geq 0$ and $\delta_1 + \delta_4 + 1 \geq 0$ and $\delta_2 + \delta_4 + 1 < 0$;

(v). $\{E_1, E_2, D_3, D_4\}$ if $\delta_2 + \delta_4 + 1 \geq 0$ and $\delta_3 + \delta_4 + 1 < 0$;
(vi). $\{E_1, E_2, E_3, E_4, D_2, D_3, D_4\}$ if $\delta_3 + \delta_4 + 1 \geq 0$ and $\rho(D_1)_1 > \sum_{j=2}^4 \rho(D_1)_j$;
(vii). $\{E_1, E_2, E_3, E_4, D_1, D_2, D_3, D_4\}$ if $\rho(D_1)_1 \leq \sum_{j=2}^4 \rho(D_1)_j$.

Proof: Assume Ξ is an $M(0)$-set in $\aleph_4$ with $Q_4(\Xi) \neq \phi$. We have the following six results.

(i) If $\Xi = C_\ell$ and $\boldsymbol{q} \in Q_4(C_\ell)$, then we see that $\delta_\ell = \sum_{\boldsymbol{\sigma}\in\Xi} q_{\boldsymbol{\sigma}} \geq \sum_{\boldsymbol{\sigma}\in\Xi} q_{\boldsymbol{\sigma}}(-1)^{\sigma_1} \geq \delta_1$. Hence by $\delta_1 \geq \delta_\ell$ we see $\sum_{\boldsymbol{\sigma}\in\Xi} q_{\boldsymbol{\sigma}}(-1)^{\sigma_1} = \delta_1$ and $\sigma_1 = 0$ for all $\boldsymbol{\sigma} \in S(\boldsymbol{q})$, i.e. $\boldsymbol{q} \in Q_4(C_1)$ and $Q_4(C_\ell) \subseteq Q_4(C_1)$.

(ii). If $\Xi = D_1$, then by Lemma 4.7 we see $\rho(D_1)_1 \leq \sum_{j=2}^4 \rho(D_1)_j$.

(iii). If $\Xi = E_3$ or E_4, then by Lemma 4.7 we see $\rho(\Xi)_3 + \rho(\Xi)_4 \geq 0$. If $\Xi = D_2$, then by Lemma 4.7 we see $\rho(D_2)_1 \leq \sum_{j\neq 2} \rho(D_2)_j$, and thus we also have $\rho(\Xi)_3 + \rho(\Xi)_4 \geq 0$. Clearly, we have $(\rho(\Xi)_3 - \delta_3) + (\rho(\Xi)_4 - \delta_4) \leq 1$ for $\Xi = E_3$, E_4, D_2. Hence we have $\delta_3 + \delta_4 + 1 \geq 0$ if $\Xi = E_3$ or E_4 or D_2.

(iv). If $\Xi = E_2$ or D_3, similar to (iii) we can get $\delta_2 + \delta_4 + 1 \geq 0$.

(v). If $\Xi = D_4$, similar to (iii) we can get $\delta_2 + \delta_3 + 1 \geq 0$.

(vi). If $\Xi = E_1$, similar to (iii) we can get $\delta_1 + \delta_4 + 1 \geq 0$. From Lemma 4.7 we also see $\sum_{j=1}^4 \rho(E_1)_j \geq 0$.

By these six results and (4.24), we can show easily that (4.54) holds. □

If both of (2.11) and (4.8) are valid, from (4.12) and Theorem 4.6, we see easily that to solve the IPP $\mathcal{P}(Q_4)$ it is sufficient only to solve the IPP $\mathcal{P}(Q_4(\Xi))$ for the $M(0)$-sets Ξ in $\{C_1\} \cup \aleph^*$, i.e. we have the following simple corollary.

Corollary 4.3: If the received N-tuple $\boldsymbol{r}$ and the 4-tuple $\boldsymbol{\delta}$ satisfy (2.11) and (4.8) respectively, then we have

$$\underline{L}[V^N_{d_1,d_2,d_3,d_4}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4)] = \min_{\Xi\in\{C_1\}\cup\aleph^*} \underline{L}'[Q_4(\Xi)]. \tag{4.55}$$

For each $M(\boldsymbol{0})$-set $\Xi \in \{C_1, D_1, D_2, D_3, D_4, E_1, E_2, E_3, E_4\}$, we give a Sub-algorithm-$Q(\Xi)$ to solve the IPP $\mathcal{P}(Q_4(\Xi))$ in the subsections 4.5.1 to 4.5.3. Using these sub-algorithms we can solve the IPP $\mathcal{P}(Q_4)$ by the following algorithm.

Algorithm for solving the IPP $\mathcal{P}(Q_4)$
Input. The N-tuple $\boldsymbol{r}$ with (2.11). The 4-tuple $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ with (4.8). $\mathcal{D}_{\boldsymbol{\sigma}}$ and $n_{\boldsymbol{\sigma}}$ defined by (4.1) for all $\boldsymbol{\sigma} \in B^4$.

Output. $\underline{L}'[Q_4] = \underline{L}[V^N_{d_1,d_2,d_3,d_4}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4)]$.
Step 1. By the definitions, determine the 4-tuples $\rho(D_1)$ and $\rho(E_1)$. And then generate the set $\aleph^*$ by Theorem 4.6, and solve the IPP $\mathcal{P}(Q_4(\Xi))$ for each $\Xi \in \{C_1\} \cup \aleph^*$ by the Sub-algorithm-$Q_4(\Xi)$.
Step 2. Output $\min_{\Xi \in \{C_1\} \cup \aleph^*} \underline{L}'[Q_4(\Xi)]$ and END. □

Since the number of operations of additions and comparisons of real numbers is of order δ_1^2 for any Sub-algorithm-$Q_4(\Xi)$, we see that the number of operations of additions and comparisons of real numbers of above algorithm is also of order δ_1^2, and thus is of order N^2.

Now we give some notations which will be used in the derivation of the Sub-algorithm-$Q_4(\Xi)$'s. For any 2^4-tuple $\boldsymbol{q} \in Q_4$ and two sequences $\boldsymbol{\sigma}', \boldsymbol{\sigma}'' \in B^4$, let $\varphi(\boldsymbol{q}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')$ denote the 2^4-tuple which satisfies

$$\begin{cases} \varphi(\boldsymbol{q}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')_{\boldsymbol{\sigma}'} = q_{\boldsymbol{\sigma}'} + 1, \varphi(\boldsymbol{q}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')_{\boldsymbol{\sigma}''} = q_{\boldsymbol{\sigma}''} - 1, \\ \varphi(\boldsymbol{q}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')_{\boldsymbol{\sigma}} = q_{\boldsymbol{\sigma}}, \text{ for all } \boldsymbol{\sigma} \in B^4 \backslash \{\boldsymbol{\sigma}', \boldsymbol{\sigma}''\}. \end{cases} \tag{4.56}$$

For any integer k with $0 \leq k \leq 15$, let $\boldsymbol{\sigma}(k)$ denote the sequence in B^4 which satisfies $\sum_{j=1}^4 \sigma(k)_j 2^{j-1} = k$. In this section, let ℓ be a fixed integer with $1 \leq \ell \leq 4$. Let $s_0 = 0000$ and $s_0' = \boldsymbol{\sigma}(2^{\ell-1})$. For $i = 1, 2, 3$, let

$$I_i \triangleq \{1,2,3\} \backslash \{i\}, \; a_i \triangleq \begin{cases} i + \ell, & \text{if } i + \ell \leq 4, \\ i + \ell - 4, & \text{otherwise,} \end{cases} \tag{4.57}$$

$$s_i \triangleq \boldsymbol{\sigma}(2^{a_i - 1}), \; s_i' \triangleq \boldsymbol{\sigma}(2^{a_i-1} + 2^{\ell-1}), \; s_i^* \triangleq \boldsymbol{\sigma}(15 - 2^{a_i-1} - 2^{\ell-1}). \tag{4.58}$$

4.5.1 Sub-algorithm for Solving the Sub-IPP $\mathcal{P}(Q_4(C_1))$

Let $C_1' \triangleq C_1 \backslash \{0000, 0001\}$ and

$$\delta_i^C \triangleq \max\{0, \lfloor (\delta_1 + \delta_i + 1)/2 \rfloor\}, \text{ for } i = 2,3,4. \tag{4.59}$$

For integers d, k with $0 \leq d \leq \delta_1$, $0 \leq k < \delta_4^C$, let $R^{d,k}$ be the set of the 2^h-tuples $\boldsymbol{q}$ in Q_4 which satisfy (4.2) and

$$\begin{cases} q_{0000} + q_{0001} = d, \; \sum_{\boldsymbol{\sigma} \in C_1'} q_{\boldsymbol{\sigma}} = \delta_1 - d, \\ q_{0010} + q_{0011} \geq \delta_2^C - d, \; q_{0100} + q_{0101} \geq \delta_3^C - d, \\ q_{0000} + q_{0010} + q_{0100} + q_{0110} \geq k, \\ q_{\boldsymbol{\sigma}} = 0, \text{ for } \boldsymbol{\sigma} \notin C_1. \end{cases} \tag{4.60}$$

Then we have $R^{d,k} \subseteq R^{d,k-1} \subseteq \cdots \subseteq R^{d,0} \subseteq Q_4^*$. Let

$$d^{\mathrm{r}} \triangleq \min\{\delta_1, n_{0000} + n_{0001}\}, \tag{4.61}$$

$$d^{\mathrm{l}} \triangleq \max\{0,\ \delta_2^{\mathrm{C}} + \delta_3^{\mathrm{C}} - \delta_1, \delta_1 - \sum_{\sigma \in C_1'} n_{\sigma}, \\ \delta_2^{\mathrm{C}} - n_{0011} - n_{0010},\ \delta_3^{\mathrm{C}} - n_{0101} - n_{0100}\}. \tag{4.62}$$

Then $R^{d,0}$ is not empty if and only if d satisfies $d^{\mathrm{l}} \leq d \leq d^{\mathrm{r}}$. From the definition of $Q_4(C_1)$ we can show easily

$$Q_4(C_1) = \bigcup_{d^{\mathrm{l}} \leq d \leq d^{\mathrm{r}}} R^{d,\delta_4^{\mathrm{C}}}. \tag{4.63}$$

Clearly, $Q_4(C_1)$-optimum can be obtained from the $R^{d,\delta_4^{\mathrm{C}}}$-optimums. Next, we give an iterative method for generating the $R^{d,\delta_4^{\mathrm{C}}}$-optimums. For d with $d^{\mathrm{l}} \leq d \leq d^{\mathrm{r}}$, let

$$v(d) \triangleq \delta_1 - d - \max\{\delta_2^{\mathrm{C}} - d, 0\} - \max\{\delta_3^{\mathrm{C}} - d, 0\} \geq 0, \tag{4.64}$$

$$V(d) \triangleq (\mathcal{D}_{0011} \cup \mathcal{D}_{0010})^{(\delta_2^{\mathrm{C}}-d)} \cup (\mathcal{D}_{0101} \cup \mathcal{D}_{0100})^{(\delta_3^{\mathrm{C}}-d)}, \tag{4.65}$$

$$V^*(d) \triangleq V(d) \cup (\mathcal{D}_{0000} \cup \mathcal{D}_{0001})^{(d)} \cup ((\cup_{\sigma \in C_1'} \mathcal{D}_{\sigma}) \backslash V(d))^{(v(d))}, \tag{4.66}$$

and let $q^{d,0}$ denote the 2^4-tuple which satisfies $\mathcal{D}(q^{d,0}) = V^*(d)$. Then $q^{d,0}$ is an $R^{d,0}$-optimum. If $R^{d,\delta_4^{\mathrm{C}}} \neq \phi$, we can find an $R^{d,\delta_4^{\mathrm{C}}}$-optimum from $q^{d,0}$ by iteration. Let $\mathcal{A} \triangleq \{(0000, 0001)\} \cup (\{0010, 0100, 0110\} \times \{0011, 0101, 0111\})$.

Lemma 4.8: Assume $q^{d,k}$ is an $R^{d,k}$-optimum which satisfies

$$q_{0000}^{d,k} + q_{0010}^{d,k} + q_{0100}^{d,k} + q_{0110}^{d,k} = k. \tag{4.67}$$

Let $\xi(q^{d,k}) \triangleq \{\varphi(q^{d,k}, \sigma', \sigma'') : (\sigma', \sigma'') \in \mathcal{A}\} \cap R^{d,k+1}$. Then $R^{d,k+1} = \phi$ if and only if $\xi(q^{d,k}) = \phi$, and the $\xi(q^{d,k})$-optimum must be an $R^{d,k+1}$-optimum if $\xi(q^{d,k}) \neq \phi$.

Proof: See Appendix F. □

Assume $q^{d,k}$ is an $R^{d,k}$-optimum which satisfies (4.67) and $\xi(q^{d,k}) \neq \phi$. Since between $L'(\varphi(q^{d,k}, \sigma, \sigma'))$ and $L'(\varphi(q^{d,k}, \beta, \beta'))$ with $\sigma = \beta$ or $\sigma' = \beta'$ we can find the smaller one with no operations of real numbers, we see easily that to determine the $\xi(q^{d,k})$-optimum it is enough to consider $L'(q) - L'(q^{d,k})$ for at most 4 tuples q of $\xi(q^{d,k})$ and thus with at most 7 operations of additions and comparisons of real numbers we can find an $R^{d,k+1}$-optimum from $q^{d,k}$. Hence, by iteration from $q^{d,0}$ with at most $7\delta_4^C$ operations of additions and comparisons of real numbers we can find an R^{d,δ_4^C}-optimum, or determine $R^{d,\delta_4^C} = \phi$. With respect to (4.63) and the definition of the R^{d,δ_4^C}-optimums, we can give the following Sub-algorithm-$Q_4(C_1)$ to compute $\underline{L}'[Q_4(C_1)]$.

Sub-algorithm-$Q_4(C_1)$
Input. The N-tuple r with (2.11). The 4-tuple $\delta = (\delta_1, \delta_2, \delta_3, \delta_4)$ with (4.8). $\mathcal{D}_\sigma$ and n_σ defined by (4.1) for all $\sigma \in C_1$.
Output. $\underline{L}'[Q_4(C_1)]$.
Step 1. Compute δ_j^C by (4.59) for $j = 2, 3, 4$, d^l by (4.61) and d^r by (4.62).
Step 2. If $d^l > d^r$, then output $+\infty$ (i.e. $Q(C_1) = \phi$) and END, otherwise, for each integer d with $d^l \le d \le d^r$, generate the set $V^*(d)$ by (4.66), and determine the 2^4-tuple $q^{d,0}$ with $\mathcal{D}(q^{d,0}) = V^*(d)$, and then, by using of Lemma 4.8, find an R^{d,δ_4^C}-optimum q^{d,δ_4^C}, or determine $R^{d,\delta_4^C} = \phi$.
Step 3. Output $\min_{d^l \le d \le d^r} L'(q^{d,\delta_4^C})$ and END. □

The number of d with $d^l \le d \le d^r$ is at most $\delta_1 + 1$. For each d with $d^l \le d \le d^r$, it needs at most $7\delta_4^C \le 7\delta_1$ operations of additions and comparisons of real numbers to find an R^{d,δ_4^C}-optimum q^{d,δ_4^C} or to determine $R^{d,\delta_4^C} = \phi$. To compute $L'(q^{d,\delta_4^C})$ it needs $\delta_1 - 1$ additions of real numbers. Hence the total number of operations of additions and comparisons of real numbers for Sub-algorithm-$Q_4(C_1)$ is not more than $8\delta_1(\delta_1 + 1)$.

4.5.2 Sub-algorithm for Solving the Sub-IPP $\mathcal{P}(Q_4(D_\ell))$

Clearly, $Q_4(D_\ell)$ consists of the tuples q in Q_4 which satisfy (4.2) and

$$\begin{cases} \sum_{j=0}^{3}(q_{s_j} + q_{s'_j}) - 2(q_{s_i} + q_{s'_i}) = \rho(D_\ell)_{a_i}, \text{ for } i = 1, 2, 3, \\ \sum_{j=0}^{3}(q_{s_j} - q_{s'_j}) \ge \delta_\ell, \text{ and } q_\sigma = 0 \text{ for } \sigma \notin D_\ell. \end{cases} \tag{4.68}$$

Let

$$\delta_i^D \triangleq \sum_{j \in I_i} \rho(D_\ell)_{a_j}/2 = \lceil \sum_{j \in I_i} \delta_{a_j}/2 \rceil, \text{ for } i = 1, 2, 3, \tag{4.69}$$

$$\delta^{\mathrm{D}} \triangleq \max\{0, \lceil(\delta_\ell + \sum_{j=1}^{3} \rho(D_\ell)_{a_j})/2\rceil\} = \max\{0, \lceil(\delta_\ell + \sum_{j=1}^{3} \delta_j^{\mathrm{D}})/2\rceil\}. \quad (4.70)$$

For nonnegative integers d, k, let $\dot{R}^{d,k}$ denote the set of the 2^h-tuples $\boldsymbol{q}$ in Q_4^* which satisfy (4.2) and

$$\begin{cases} q_{s_0} + q_{s'_0} = d, \ \sum_{j=0}^{3} q_{s_j} \geq k, \\ q_{s_i} + q_{s'_i} = \delta_i^{\mathrm{D}} - d, \ \text{for } i = 1, 2, 3, \\ q_{\boldsymbol{\sigma}} = 0, \ \text{for } \boldsymbol{\sigma} \notin D_\ell. \end{cases} \quad (4.71)$$

Clearly, we have $\dot{R}^{d,k} \subseteq \dot{R}^{d,k-1} \subseteq \cdots \subseteq \dot{R}^{d,0} \subseteq Q_4^*$. Let

$$\dot{d}^{\mathrm{l}} \triangleq \max\{0, \max_{1\leq i\leq 3}(\delta_i^{\mathrm{D}} - n_{s_i} - n_{s'_i})\}, \quad (4.72)$$

$$\dot{d}^{\mathrm{r}} \triangleq \min\{n_{s_0} + n_{s'_0}, \ \min_{1\leq i\leq 3} \delta_i^{\mathrm{D}}\}. \quad (4.73)$$

Then from (4.2) and (4.68) to (4.73) we can prove easily that

$$Q_4(D_\ell) = \bigcup_{\dot{d}^{\mathrm{l}} \leq d \leq \dot{d}^{\mathrm{r}}} \dot{R}^{d,\delta^{\mathrm{D}}-d}. \quad (4.74)$$

For d with $\dot{d}^{\mathrm{l}} \leq d \leq \dot{d}^{\mathrm{r}}$, write

$$W(d) \triangleq (\mathcal{D}_{s_0} \cup \mathcal{D}_{s'_0})^{(d)} \cup \bigcup_{1\leq i\leq 3} (\mathcal{D}_{s_i} \cup \mathcal{D}_{s'_i})^{(\delta_i^{\mathrm{D}}-d)}, \quad (4.75)$$

and let $\dot{\boldsymbol{q}}^{d,0}$ denote the 2^4-tuple which satisfies $\mathcal{D}(\dot{\boldsymbol{q}}^{d,0}) = W(d)$. Then $\dot{\boldsymbol{q}}^{d,0}$ must be an $\dot{R}^{d,0}$-optimum. If $\dot{R}^{d,\delta^{\mathrm{D}}-d} \neq \phi$, the following lemma suggests an iterative method for finding an $\dot{R}^{d,\delta^{\mathrm{D}}-d}$-optimum from $\dot{\boldsymbol{q}}^{d,0}$ too.

Lemma 4.9: Assume $\dot{\boldsymbol{q}}^{d,k}$ is an $\dot{R}^{d,k}$-optimum which satisfies

$$\dot{q}_{s_0}^{d,k} + \dot{q}_{s_1}^{d,k} + \dot{q}_{s_2}^{d,k} + \dot{q}_{s_3}^{d,k} = k. \quad (4.76)$$

Let $\dot{\xi}(\dot{\boldsymbol{q}}^{d,k}) \triangleq \{\varphi(\dot{\boldsymbol{q}}^{d,k}, s_j, s'_j) : 0 \leq j \leq 3\} \cap \dot{R}^{d,k+1}$. Then $\dot{R}^{d,k+1} = \phi$ if and only if $\dot{\xi}(\dot{\boldsymbol{q}}^{d,k}) = \phi$, and the $\dot{\xi}(\dot{\boldsymbol{q}}^{d,k})$-optimum is an $\dot{R}^{d,k+1}$-optimum if $\dot{\xi}(\dot{\boldsymbol{q}}^{d,k}) \neq \phi$.

Proof. This lemma is an analogue of Lemma 4.8. We omit the proof here. □

Similar to the Sub-algorithm-$Q_4(C_1)$, we can found a Sub-algorithm-$Q_4(D_\ell)$ to compute $\underline{L}'[Q_4(D_\ell)]$.

Sub-algorithm-$Q_4(D_\ell)$

Input. The N-tuple $\boldsymbol{r}$ with (2.11). The 4-tuple $\delta = (\delta_1, \delta_2, \delta_3, \delta_4)$ with (4.8). $\mathcal{D}_\alpha$ and n_α defined by (4.1) for all $\alpha \in D_\ell$.

Output. $\underline{L}'[Q_4(D_\ell)]$.

Step 1. Determine the integers a_i for $i = 1, 2, 3$, the 4-tuples s_i and s_i' for $i = 0, 1, 2, 3$. Compute the integers δ_i^{P} by (4.69) for $i = 1, 2, 3$, δ^{D} by (4.70), $\dot{d}^{\mathrm{l}}$ by (4.72) and $\dot{d}^{\mathrm{r}}$ and by (4.73).

Step 2. If $\dot{d}^{\mathrm{l}} > \dot{d}^{\mathrm{r}}$, then output $+\infty$ (i.e. $Q_4(D_\ell) = \emptyset$) and END. If $\dot{d}^{\mathrm{l}} \leq \dot{d}^{\mathrm{r}}$, for each integer d with $\dot{d}^{\mathrm{l}} \leq d \leq \dot{d}^{\mathrm{r}}$, generate the set $W(d)$ by (4.75) and determine the 2^4-tuple $\dot{\boldsymbol{q}}^{d,0}$ with $\mathcal{D}(\dot{\boldsymbol{q}}^{d,0}) = W(d)$. And then, by using of Lemma 4.9, find an $\dot{R}^{d,\delta^{\mathrm{D}}-d}$-optimum $\dot{\boldsymbol{q}}^{d,\delta^{\mathrm{D}}-d}$, or determine $R^{d,\delta^{\mathrm{D}}-d} = \emptyset$.

Step 3. Output $\min_{\dot{d}^{\mathrm{l}} \leq d \leq \dot{d}^{\mathrm{r}}} L'(\dot{\boldsymbol{q}}^{d,\delta^{\mathrm{D}}-d})$ and END. □

The number of operations of additions and comparisons of real numbers for Sub-algorithm-$Q_4(D_\ell)$ is not more than $\sum_{d=0}^{\delta_1}(7(2\delta_1 - d) + 3\delta_1 - 2d) = 25\delta_1(\delta_1 + 1)/2$.

4.5.3 Sub-algorithm for Solving the Sub-IPP $\mathcal{P}(Q_4(E_\ell))$

Clearly, $Q_4(E_\ell)$ consists of the 2^4-tuples $\boldsymbol{q}$ in Q_4 which satisfy (4.2) and

$$\begin{cases} q_{s_0} + q_{s_0'} + \sum_{j \in I_i}(q_{s_j} - q_{s_j^*}) - q_{s_i} + q_{s_i^*} = \rho(E_\ell)_{a_i}, \text{ for } i = 1, 2, 3, \\ q_{s_0} - q_{s_0'} + \sum_{j=1}^{3}(q_{s_j} + q_{s_j^*}) = \rho(E_\ell)_\ell, \text{ and } q_{\boldsymbol{\sigma}} = 0 \text{ for } \boldsymbol{\sigma} \notin E_\ell. \end{cases} \tag{4.77}$$

Let

$$\delta_i^{\mathrm{E}} \triangleq (\rho(E_\ell)_\ell + \rho(E_\ell)_{a_i})/2 = \lceil(\delta_\ell + \delta_{a_i})/2\rceil, \text{ for } i = 1, 2, 3, \tag{4.78}$$

$$\delta^{\mathrm{E}} \triangleq \sum_{j=1}^{4} \rho(E_\ell)_j/2 = \begin{cases} \lceil\sum_{j=1}^{4} \delta_j/2\rceil, & \text{if } \sum_{j=1}^{4} \delta_j \text{ is odd}, \\ \sum_{j=1}^{3} \delta_j^{\mathrm{E}} - \delta_\ell, & \text{otherwise}. \end{cases} \tag{4.79}$$

Then from (4.2) and (4.77) to (4.79) we can show easily that the 2^4-tuples $\boldsymbol{q}$ in $Q_4(E_\ell)$ can be given by

$$\begin{cases} q_{s_0'} = \delta^{\mathrm{E}} - w, \ q_{s_0} = 2w - x_1 - x_2 - x_3, \\ q_{s_i^*} = \delta_i^{\mathrm{E}} - x_i, \ q_{s_i} = \sum_{j \in I_i} x_j - w, \text{ for } i = 1, 2, 3, \\ q_{\boldsymbol{\sigma}} = 0, \text{ for } \boldsymbol{\sigma} \notin E_\ell. \end{cases} \tag{4.80}$$

where w, x_1, x_2, x_3 satisfy

$$\max\{0, \delta^{\mathrm{E}} - n_{s'_0}\} \le w \le \delta^{\mathrm{E}}, \tag{4.81}$$

$$0 \le 2w - x_1 - x_2 - x_3 \le n_{s_0}, \tag{4.82}$$

$$\max\{0, \delta_i^{\mathrm{E}} - n_{s_i^*}\} \le x_i \le \delta_i^{\mathrm{E}},\ 0 \le \sum_{j \in I_i} x_j - w \le n_{s_i}, \text{ for } i = 1, 2, 3. \tag{4.83}$$

For any integer w with (4.81), let $\Omega(w)$ denote the set of pairs $\pi \triangleq (x_1, x_2, x_3)$ which satisfy (4.82) and (4.83). For $\pi \in \Omega(w)$, let $\boldsymbol{q}_w(\pi)$ be the 2^4-tuple of $Q(E_\ell)$ defined by (4.80). We write $L_w(\pi) \triangleq L'(\boldsymbol{q}_w(\pi))$ for $\pi \in \Omega(w)$ and $L_w(\pi) \triangleq +\infty$ for $\pi \notin \Omega(w)$. For any subset Ω' of $\Omega(w)$, let $L_w[\Omega'] \triangleq \min_{\pi \in \Omega'} L_w(\pi)$, and write $L_w[\phi] \triangleq +\infty$. If a pair $\pi \in \Omega'$ satisfies $L_w(\pi) = L_w[\Omega']$, we call it **Ω'-pair**. We will show a method to evaluate $\underline{L}'[Q(E_\ell)]$ by finding an $\Omega(w)$-pair for each w with $\Omega(w) \neq \phi$.

For integers w, x, let $\Omega(w, x) \triangleq \{\pi \in \Omega(w) : \pi = (\cdot, \cdot, x)\}$. At first, we consider to give an $\Omega(w, x)$-pair for nonempty set $\Omega(w, x)$. Let Υ denote the set of pairs $\pi' = (x'_1, x'_2, x'_3)$ with $x'_i \in \{1, 0, -1\}$. For $\pi \in \Omega(w)$ and $\pi' \in \Upsilon$, we say that we can **grow π in the π'-direction** to $\pi + \pi'$ if $L_w(\pi) > L_w(\pi + \pi')$. For any nonempty set $\Omega(w, x)$, we can find an $\Omega(w, x)$-pair by the following growth procedure.

Growth Procedure of an $\Omega(w, x)$-pair $\tau(w, x)$
Preparation. Select an arbitrary pair π of $\Omega(w, x)$ as the seed.
Step 1. We consider to grow π in the $(1, 0, 0)$-direction and $(-1, 0, 0)$-direction till we can not do anymore and then goto Step 2.
Step 2. If we can grow π in π'-direction for some π' of $\{(1, 0, 0), (-1, 0, 0), (0, 1, 0), (0, -1, 0)\}$, we grow it in π'-direction step by step till we can not do anymore and then goto Step 2, otherwise goto Step 3.
Step 3. We consider to grow π in $(1, -1, 0)$-direction and $(-1, 1, 0)$-direction till we can not do anymore and then output $\tau(w, x) = \pi$ and END. □

We have the following lemma.

Lemma 4.10: The output $\tau(w, x)$ of the above growth procedure is an $\Omega(w, x)$-pair and the growth route must be one of the four classes shown in Fig. 4.1.

Proof: See Appendix G. □

Figure 4.1 The possible routes of the growth of $\tau(w, z)$

Since for each step of growth or change of direction it needs 4 additions and comparisons of real numbers, according to Fig. 4.1 we can show easily that from an arbitrary pair π of $\Omega(w, x)$ with at most $16w$ operations of additions and comparisons of real numbers we can find an $\Omega(w, x)$-pair by the above growth procedure.

Sometimes it is not easy to select a seed for this growth procedure. Below we give a lemma which suggests a concrete method for giving a pair π from any nonempty set $\Omega(w, x)$.

Lemma 4.11: (i) $\Omega(w) \neq \phi$ if and only if

$$0 \leq \delta_1^{\mathrm{E}} \leq n_{s_0} + n_{s_2} + n_{s_3} + n_{s_1^*}, \tag{4.84}$$

$$0 \leq \delta_2^{\mathrm{E}} \leq n_{s_0} + n_{s_1} + n_{s_3} + n_{s_2^*}, \tag{4.85}$$

$$0 \leq \delta_3^{\mathrm{E}} \leq n_{s_0} + n_{s_1} + n_{s_2} + n_{s_3^*}, \tag{4.86}$$

$$\max\{\delta_1^{\mathrm{E}} - n_{s_1^*}, \delta_2^{\mathrm{E}} - n_{s_2^*}, \delta_3^{\mathrm{E}} - n_{s_3^*}\} \leq \min\{\delta_1^{\mathrm{E}} + n_{s_1}, \delta_2^{\mathrm{E}} + n_{s_2}, \delta_3^{\mathrm{E}} + n_{s_3}\}, \tag{4.87}$$

and $w^{\mathrm{l}} \leq w \leq w^{\mathrm{r}}$, where

$$\begin{aligned} w^{\mathrm{r}} \triangleq \min\{&\delta^{\mathrm{E}}, n_{s_1} + n_{s_2} + n_{s_3} + 2n_{s_0}, \lfloor \frac{\delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}} + n_{s_0}}{2} \rfloor, \\ &2\delta_1^{\mathrm{E}} + n_{s_1}, 2\delta_2^{\mathrm{E}} + n_{s_2}, 2\delta_3^{\mathrm{E}} + n_{s_3}, \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}}, \delta_1^{\mathrm{E}} + \delta_3^{\mathrm{E}}, \delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}}, \\ &\delta_1^{\mathrm{E}} + n_{s_1} + n_{s_0}, \delta_2^{\mathrm{E}} + n_{s_2} + n_{s_0}, \delta_3^{\mathrm{E}} + n_{s_3} + n_{s_0}\}, \end{aligned} \tag{4.88}$$

$$\begin{aligned} w^{\mathrm{l}} \triangleq \max\{&0, \delta^{\mathrm{E}} - n_{s_0'}, \lceil \frac{\delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}} - n_{s_1^*} - n_{s_2^*} - n_{s_3^*}}{2} \rceil, \\ &\delta_1^{\mathrm{E}} - n_{s_1^*}, 2\delta_1^{\mathrm{E}} - 2n_{s_1^*} - n_{s_2} - n_{s_3}, \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}} - n_{s_1} - n_{s_2^*} - n_{s_3^*}, \\ &\delta_2^{\mathrm{E}} - n_{s_2^*}, 2\delta_2^{\mathrm{E}} - 2n_{s_2^*} - n_{s_1} - n_{s_3}, \delta_1^{\mathrm{E}} + \delta_3^{\mathrm{E}} - n_{s_2} - n_{s_1^*} - n_{s_3^*}, \\ &\delta_3^{\mathrm{E}} - n_{s_3^*}, 2\delta_3^{\mathrm{E}} - 2n_{s_3^*} - n_{s_1} - n_{s_2}, \delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}} - n_{s_3} - n_{s_1^*} - n_{s_2^*}\}. \end{aligned} \tag{4.89}$$

(ii) Assume $\Omega(w) \neq \phi$. $\Omega(w,z) \neq \phi$ if and only if $z^{l}(w) \leq z \leq z^{r}(w)$, where

$$\begin{aligned} z^{r}(w) \triangleq \min\{w, \delta_3^{E}, \lfloor\frac{w+n_{s1}+n_{s2}}{2}\rfloor, \delta_1^{E}+n_{s1}, \delta_2^{E}+n_{s2}, \\ n_{s1}+n_{s2}+n_{s0}, 2w-\delta_1^{E}-\delta_2^{E}+n_{s_1^*}+n_{s_2^*}, \\ w-\delta_1^{E}+n_{s_1^*}+n_{s2}, w-\delta_2^{E}+n_{s_2^*}+n_{s1}\}, \end{aligned} \tag{4.90}$$

$$\begin{aligned} z^{l}(w) \triangleq \max\{0, \lceil\frac{w-n_{s3}}{2}\rceil, w-n_{s3}-n_{s0}, 2w-\delta_1^{E}-\delta_2^{E}-n_{s0}, \\ \delta_3^{E}-n_{s_3^*}, w-\delta_1^{E}, w-\delta_2^{E}, \delta_1^{E}-n_{s_1^*}-n_{s3}, \delta_2^{E}-n_{s_2^*}-n_{s3}\}. \end{aligned} \tag{4.91}$$

(iii) If $\Omega(w,z) \neq \phi$, then $\Omega(w,z)$ contains at least one of the following four pairs:

$$(x^{l}_{w,z},\ \sigma_{w,z}-x^{l}_{w,z},\ z),\ (x^{r}_{w,z},\ \sigma_{w,z}-x^{r}_{w,z},\ z), \tag{4.92}$$

$$(\sigma_{w,z}-y^{l}_{w,z},\ y^{l}_{w,z},\ z),\ (\sigma_{w,z}-y^{r}_{w,z},\ y^{r}_{w,z},\ z), \tag{4.93}$$

where

$$x^{l}_{w,z} \triangleq \max\{0,\ \delta_1^{E}-n_{s_1^*},\ w-z\},\ x^{r}_{w,z} \triangleq \min\{\delta_1^{E},\ n_{s2}+w-z\}, \tag{4.94}$$

$$y^{l}_{w,z} \triangleq \max\{0,\ \delta_2^{E}-n_{s_2^*},\ w-z\},\ y^{r}_{w,z} \triangleq \min\{\delta_2^{E},\ n_{s1}+w-z\}, \tag{4.95}$$

$$\sigma_{w,z} \triangleq \max\{w, 2w-z-n_{s0}, x^{l}_{w,z}+y^{l}_{w,z}\}. \tag{4.96}$$

□

However, it is not needed to find all the $\Omega(w,x)$-pairs by the growth procedure. Indeed, the following lemma suggests a simple method for finding a $\Omega(w)$-pair from a known $\Omega(w,x)$-pair.

Lemma 4.12: Let π be a $\Omega(w,x)$-pair and

$$\Upsilon^* \triangleq \{(0,0,0),(-1,0,0),(0,-1,0),(1,-1,0),(-1,1,0),(-1,-1,0)\}. \tag{4.97}$$

1. If $\Omega(w,x+1) \neq \phi$, there exists a pair π' in Υ^* such that $\pi+(0,0,1)+\pi'$ is a $\Omega(w,x+1)$-pair.
2. If $\Omega(w,x-1) \neq \phi$, there exists a pair π'' in Υ^* such that $\pi-(0,0,1)-\pi''$ is a $\Omega(w,x-1)$-pair.
3. π is a $\Omega(w)$-pair if and only if

$$L_w(\pi) \leq \min\{L_w[\Omega(w,x-1)], L_w[\Omega(w,x+1)]\}. \tag{4.98}$$

Proof: See Appendix H. □

We can use the following Sub-algorithm-$Q_4(E_\ell)$ to evaluate $\underline{L}'[Q_4(E_\ell)]$.

Sub-algorithm-$Q_4(E_\ell)$

Input. The N-tuple $\boldsymbol{r}$ with (2.11). The 4-tuple $\delta = (\delta_1, \delta_2, \delta_3, \delta_4)$ with (4.8). $\mathcal{D}_\alpha$ and n_α defined by (4.1) for all $\alpha \in E_\ell$.

Output. $\underline{L}'[Q_4(E_\ell)]$.

Step 1. Determine the integers a_1, a_2, a_3, and the 4-tuples s_0, s_1, s_2, s_3, s_0', s_1^*, s_2^*, s_3^*. Compute the integers δ_i^{E} by (4.78) for $i = 1, 2, 3$, and δ^{E} by (4.79). Then use (4.88) and (4.89) to compute w^{r} and w^{l}.

Step 2. If one of (4.84)-(4.87) or $w^{\mathrm{l}} \leq w^{\mathrm{r}}$ is not valid, then output $+\infty$ (i.e. $Q_4(E_\ell) = \emptyset$) and END, otherwise, for each integer w with $w^{\mathrm{l}} \leq w \leq w^{\mathrm{r}}$, by (4.90) and (4.91) compute $z^{\mathrm{r}}(w)$ and $z^{\mathrm{l}}(w)$. Let $z_w = \lfloor (z^{\mathrm{r}}(w) + z^{\mathrm{l}}(w))/2 \rfloor$. Select a pair in $\Omega(w, z_w)$ by (4.92) and (4.93), and then generate an $\Omega(w, z_w)$-pair $\tau(w, z_w)$ by the growth procedure. Then according to Lemma 4.12 to find an $\Omega(w)$-pair $\tau(w)$.

Step 3. Output $\min_{w^{\mathrm{l}} \leq w \leq w^{\mathrm{r}}} L_w(\tau(w))$ and END. □

We consider the computational complexity of this sub-algorithm. According to Lemma 4.12, from an $\Omega(w, x)$-pair $\tau(w, x)$ we can get an $\Omega(w, x+1)$-pair $\tau(w, x+1)$ (or an $\Omega(w, x-1)$-pair $\tau(w, x-1)$) with at most 20 operations of additions and comparisons of real numbers, and determine whether $L_w(\tau(w, x)) \leq L_w(\tau(w, x+1))$ (or $L_w(\tau(w, x)) \leq L_w(\tau(w, x-1))$) holds or not with 5 more operations of additions and comparisons of real numbers. Since $\Omega(w, x) = \phi$ if $x > w$, we can find an $\Omega(w)$-pair $\tau(w)$ with at most $25w/2$ operations of additions and comparisons of real numbers. Furthermore, since the number of additions and comparisons of real numbers for computing $L_w(\tau(w))$ is $\delta^{\mathrm{E}} + \delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}} - 2w - 1$ and $\Omega(w) = \phi$ for $w > \delta^{\mathrm{E}}$, the number of additions and comparisons of real numbers for evaluating $\underline{L}'[Q_4(E_\ell)]$ with Sub-algorithm-$Q_4(E_\ell)$ is not more than $\sum_{w=0}^{\delta^{\mathrm{E}}} (53w/2 + \delta^{\mathrm{E}} + \delta_1^{\mathrm{E}} + \delta_2^{\mathrm{E}} + \delta_3^{\mathrm{E}}) \leq 63\delta_1(2\delta_1 + 1)/2$.

4.6 An Improvement of the Optimality Condition

We notice that the set $V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)$ in Lemma 2.1 can be replaced with any superset of $C\setminus(\bigcup_{j=1}^h O_{d_j-1}(\boldsymbol{u}_j))$. If some of the reference words $\boldsymbol{u}_j$ are codewords, and the distance profile $W_C \triangleq \{0, w_1 = d_{\min}, w_2, \ldots,\}$ or its partial knowledge is known, then we can design some stronger Cond_{opt}s.

For any integer d and 2^N-tuple $\boldsymbol{v} \in V^N$, let

$$\phi_d(\boldsymbol{v}) \triangleq \{\boldsymbol{u} \in V^N : d_H(\boldsymbol{v},\boldsymbol{u}) = d\}. \tag{4.99}$$

Assume the first h_0 reference words are codewords. For j with $1 \le j \le h_0$, let $i(j) \triangleq \min\{i : w_i \ge d_j\}$. For any sequence $\boldsymbol{\alpha} \in B^{h_0}$, define

$$V(h,\boldsymbol{\alpha}) \triangleq \bigcap_{1\le j\le h_0,\alpha_j=0} \phi_{w_{i(j)}}(\boldsymbol{u}_j) \bigcap_{1\le j\le h_0,\alpha_j=1} \bar{O}_{w_{i(j)+1}}(\boldsymbol{u}_j) \bigcap_{h_0<j\le h} \bar{O}_{d_j}(\boldsymbol{u}_j). \tag{4.100}$$

Then the set $\bigcup_{\boldsymbol{\alpha}\in B^{h_0}} V(h,\boldsymbol{\alpha})$ is a superset of $C\setminus(\bigcup_{j=1}^h O_{d_j-1}(\boldsymbol{u}_j))$, and thus the result of Lemma 2.1 is still valid if we replace the set $V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_h)$ in Lemma 2.1 with $\bigcup_{\boldsymbol{\alpha}\in B^{h_0}} V(h,\boldsymbol{\alpha})$.

Lemma 4.13: Assume that $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, ..., $\boldsymbol{u}_{h_0}$ are codewords and $\boldsymbol{c}_{\text{best}}$ is a codeword in $\bigcup_{j=1}^h O_{d_j-1}(\boldsymbol{u}_j)$ with $h \ge h_0$. If

$$L(\boldsymbol{c}_{\text{best}}) \le \underline{L}\left[\bigcup_{\boldsymbol{\alpha}\in B^{h_0}} V(h,\boldsymbol{\alpha})\right], \tag{4.101}$$

then the optimal codeword $\boldsymbol{c}_{\text{opt}}$ must belong to the set $\bigcup_{j=1}^h O_{d_j-1}(\boldsymbol{u}_j)$.

□

It is needed to evaluate $\underline{L}[\bigcup_{\boldsymbol{\alpha}\in B^{h_0}} V(h,\boldsymbol{\alpha})]$. Clearly, we have

$$\underline{L}\left[\bigcup_{\boldsymbol{\alpha}\in B^{h_0}} V(h,\boldsymbol{\alpha})\right] = \min_{\boldsymbol{\alpha}\in B^{h_0}} \underline{L}[V(h,\boldsymbol{\alpha})] \tag{4.102}$$

Let $Q(h,\boldsymbol{\alpha})$ denote the set of all 2^h-tuples $\boldsymbol{q} \in Q_h$ which satisfy (4.2) and

$$\begin{cases} \sum_{\boldsymbol{\sigma}\in B^h} q_{\boldsymbol{\sigma}}(-1)^{\sigma_i} = \delta'_i, \text{ for } i \in [1,h_0] \text{ with } \alpha_i = 0, \\ \sum_{\boldsymbol{\sigma}\in B^h} q_{\boldsymbol{\sigma}}(-1)^{\sigma_i} \ge \delta''_i, \text{ for } i \in [1,h_0] \text{ with } \alpha_i = 1, \\ \sum_{\boldsymbol{\sigma}\in B^h} q_{\boldsymbol{\sigma}}(-1)^{\sigma_i} \ge \delta_i, \text{ for } i \in [h_0+1,h], \end{cases} \tag{4.103}$$

where

$$\delta_i' \triangleq w_{i(j)} - |\mathcal{D}_1(\boldsymbol{u}_i)|, \tag{4.104}$$

$$\delta_i'' \triangleq w_{i(j)+1} - |\mathcal{D}_1(\boldsymbol{u}_i)|. \tag{4.105}$$

Then, similar to Theorem 4.1, we have the following theorem.

Theorem 4.7: If the received sequence $\boldsymbol{r}$ satisfies (2.11), then we have

$$\underline{L}'[Q(h,\boldsymbol{\alpha})] = \underline{L}[V(h,\boldsymbol{\alpha})], \text{ for all } \boldsymbol{\alpha} \in B^{h_0}. \tag{4.106}$$

4.6.1 Evaluation of the Improved Optimality Condition for the Case $h = 1$

In this subsection, we evaluate the improved optimality condition for the case that there is only one reference word, which is also codeword. Then $h = h_0 = 1$. $Q(1,0)$ consists of the tuple $\boldsymbol{q} = (q_0, q_1)$ which satisfies

$$q_0 - q_1 = \delta_1', \ 0 \le q_0 \le n_0, \ 0 \le q_1 \le n_1. \tag{4.107}$$

Clearly, we have

$$-n_1 = -n(\boldsymbol{u}_1) \le \delta_1' \le N - n(\boldsymbol{u}_1) = n_0. \tag{4.108}$$

Hence, we have

$$\underline{L}'[Q(1,0)] = \sum_{i \in \mathcal{D}_0^{(\delta_1')} \cup \mathcal{D}_1^{(-\delta_1')}} |r_i|. \tag{4.109}$$

$Q(1,1)$ consists of the tuple $\boldsymbol{q} = (q_0, q_1)$ which satisfies

$$q_0 - q_1 \ge \delta_1'', \ 0 \le q_0 \le n_0, \ 0 \le q_1 \le n_1. \tag{4.110}$$

From $\delta_1'' \le N - n(\boldsymbol{u}_1) = n_0$, we see

$$\underline{L}'[Q(1,1)] = \sum_{i \in \mathcal{D}_0^{(\delta_1'')}} |r_i|. \tag{4.111}$$

Furthermore, we can get

$$\underline{L}'[Q(1,0) \cup Q(1,1)] = \begin{cases} \min\{\sum_{i \in \mathcal{D}_1^{(-\delta_1')}} |r_i|, \sum_{i \in \mathcal{D}_0^{(\delta_1'')}} |r_i|\}, & \text{if } \delta_1' < 0 < \delta_1'', \\ \sum_{i \in \mathcal{D}_0^{(\delta_1')}} |r_i|, & \text{otherwise.} \end{cases} \tag{4.112}$$

4.6.2 Evaluation of the Improved Optimality Condition for the Case $h = 2$

In this subsection, we evaluate the improved optimality condition for the case that there are two reference words.

Case 1: One of the two reference words is codeword, i.e. $h_0 = 1$.

Assume $\boldsymbol{q}^*$ is a minimal $Q(2,0)$-optimum. We see easily that

$$q_{11}^* q_{00}^* = q_{11}^* q_{01}^* = q_{10}^* q_{01}^* = 0. \tag{4.113}$$

If $q_{11}^* + q_{10}^* = 0$, then

$$\begin{cases} q_{00}^* + q_{01}^* = \delta_1', \; q_{00}^* - q_{01}^* \geq \delta_2, \\ 0 \leq q_{00}^* \leq n_{00}, \; 0 \leq q_{01}^* \leq n_{01}, \; q_{11}^* = q_{10}^* = 0, \end{cases}$$

i.e.

$$\begin{cases} q_{00}^* + q_{01}^* = \delta_1', \; q_{01}^* \leq \lfloor (\delta_1' - \delta_2)/2 \rfloor, \\ 0 \leq q_{00}^* \leq n_{00}, \; 0 \leq q_{01}^* \leq n_{01}, \; q_{11}^* = q_{10}^* = 0. \end{cases}$$

Hence, we have

$$\delta_1' \geq 0, \; \delta_1' - \delta_2 \geq 0, \tag{4.114}$$

$$L'(\boldsymbol{q}^*) = \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor (\delta_1' - \delta_2)/2 \rfloor)})^{(\delta_1')}} |r_i|. \tag{4.115}$$

If $q_{00}^* + q_{01}^* = 0$, then

$$\begin{cases} -q_{11}^* - q_{10}^* = \delta_1', \; -q_{11}^* + q_{10}^* \geq \delta_2, \\ 0 \leq q_{11}^* \leq n_{11}, \; 0 \leq q_{10}^* \leq n_{10}, \; q_{00}^* = q_{01}^* = 0, \end{cases}$$

i.e.

$$\begin{cases} q_{11}^* + q_{10}^* = -\delta_1', \; q_{11}^* \leq \lfloor (-\delta_1' - \delta_2)/2 \rfloor, \\ 0 \leq q_{11}^* \leq n_{11}, \; 0 \leq q_{10}^* \leq n_{10}, \; q_{00}^* = q_{01}^* = 0. \end{cases}$$

Hence, we have

$$\delta_1' \leq 0, \; \delta_1' + \delta_2 \leq 0, \tag{4.116}$$

$$L'(\boldsymbol{q}^*) = \sum_{i \in (\mathcal{D}_{10} \cup \mathcal{D}_{11}^{(\lfloor (-\delta_1' - \delta_2)/2 \rfloor)})^{(-\delta_1')}} |r_i|. \tag{4.117}$$

If $q_{00}^* + q_{01}^* > 0$ and $q_{11}^* + q_{10}^* > 0$, from (4.113) we see $q_{11}^* = 0$, $q_{10}^* > 0$, $q_{01}^* = 0$ and $q_{00}^* > 0$. If $\Delta(\boldsymbol{q}^*)_2 = q_{00}^* + q_{10}^* - q_{01}^* - q_{11}^* = q_{00}^* + q_{10}^* \geq \delta_2 + 2$, then the tuple $\boldsymbol{q} = (q_{00} = q_{00}^* - 1, q_{10} = q_{10}^* - 1, q_{01} = q_{01}^*, q_{11} = q_{11}^*)$ belongs to $Q(2,0)$, contracts the definition of $\boldsymbol{q}^*$. Hence,

$$\begin{cases} q_{00}^* - q_{10}^* = \delta_1', \ \delta_2 \leq q_{00}^* + q_{10}^* \leq \delta_2 + 1, \\ 0 \leq q_{00}^* \leq n_{00}, \ 0 \leq q_{10}^* \leq n_{10}, \ q_{11}^* = q_{01}^* = 0. \end{cases}$$

Then, we have

$$\lceil(\delta_1' + \delta_2)/2\rceil \geq 0, \ \lceil(\delta_2 - \delta_1')/2\rceil \geq 0, \tag{4.118}$$

$$L'(\boldsymbol{q}^*) = \sum_{i \in \mathcal{D}_{00}^{(\lceil(\delta_1'+\delta_2)/2\rceil)} \cup \mathcal{D}_{10}^{(\lceil(\delta_2-\delta_1')/2\rceil)}} |r_i|. \tag{4.119}$$

Furthermore, we notice that if δ_1' and δ_2 satisfy (4.114) and (4.116), or (4.114) and (4.118), or (4.116) and (4.118), then (4.115) and (4.117), or (4.115) and (4.119), or (4.117) and (4.119) are the same, respectively. Hence, we have

$$\underline{L}'[Q(2,0)] = \begin{cases} \sum_{i \in (\mathcal{D}_{10} \cup \mathcal{D}_{11}^{(\lfloor(-\delta_1'-\delta_2)/2\rfloor)})^{(-\delta_1')}} |r_i|, \text{ if } \delta_1' < 0 \text{ and } \delta_1' + \delta_2 \leq 0, \\ \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1'-\delta_2)/2\rfloor)})^{(\delta_1')}} |r_i|, \text{ if } \delta_1' \geq 0 \text{ and } \delta_1' - \delta_2 \geq 0, \\ \sum_{i \in \mathcal{D}_{00}^{(\lceil(\delta_1'+\delta_2)/2\rceil)} \cup \mathcal{D}_{10}^{(\lceil(\delta_2-\delta_1')/2\rceil)}} |r_i|, \text{ otherwise.} \end{cases} \tag{4.120}$$

Since $V(2,1)$ is a set of kind of $V_{d_1,d_2}^{\mathrm{N}}(\boldsymbol{u}_1, \boldsymbol{u}_2)$, according to the result of section 4.3, we see

$$\underline{L}'[Q(2,1)] = \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1''-\delta_2)/2\rfloor)} \cup \mathcal{D}_{10}^{(\lfloor(\delta_2-\delta_1'')/2\rfloor)})^{(\max\{\delta_1'',\delta_2\})}} |r_i|. \tag{4.121}$$

□

Case 2: Both the two reference words are codeword, i.e. $h_0 = 2$.

From (4.120), we see

$$\underline{L}'[Q(2,01)] = \begin{cases} \sum_{i \in (\mathcal{D}_{10} \cup \mathcal{D}_{11}^{(\lfloor(-\delta_1'-\delta_2'')/2\rfloor)})^{(-\delta_1')}} |r_i|, \text{ if } \delta_1' < 0 \text{ and } \delta_1' + \delta_2'' \leq 0, \\ \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1'-\delta_2'')/2\rfloor)})^{(\delta_1')}} |r_i|, \text{ if } \delta_1' \geq 0 \text{ and } \delta_1' - \delta_2'' \geq 0, \\ \sum_{i \in \mathcal{D}_{00}^{(\lceil(\delta_1'+\delta_2'')/2\rceil)} \cup \mathcal{D}_{10}^{(\lceil(\delta_2''-\delta_1')/2\rceil)}} |r_i|, \text{ otherwise.} \end{cases} \tag{4.122}$$

$$\underline{L}'[Q(2,10)] = \begin{cases} \sum_{i \in (\mathcal{D}_{01} \cup \mathcal{D}_{11}^{(\lfloor(-\delta_2'-\delta_1'')/2\rfloor)})^{(-\delta_2')}} |r_i|, \text{ if } \delta_2' < 0 \text{ and } \delta_2' + \delta_1'' \leq 0, \\ \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{10}^{(\lfloor(\delta_2'-\delta_1'')/2\rfloor)})^{(\delta_2')}} |r_i|, \text{ if } \delta_2' \geq 0 \text{ and } \delta_2' - \delta_1'' \geq 0, \\ \sum_{i \in \mathcal{D}_{00}^{(\lceil(\delta_2'+\delta_1'')/2\rceil)} \cup \mathcal{D}_{01}^{(\lceil(\delta_1''-\delta_2')/2\rceil)}} |r_i|, \text{ otherwise.} \end{cases} \tag{4.123}$$

From (4.121), we see

$$\underline{L}'[Q(2,11)] = \sum_{i \in (\mathcal{D}_{00} \cup \mathcal{D}_{01}^{(\lfloor(\delta_1''-\delta_2'')/2\rfloor)} \cup \mathcal{D}_{10}^{(\lfloor(\delta_2''-\delta_1'')/2\rfloor)})^{(\max\{\delta_1'',\delta_2''\})}} |r_i|. \tag{4.124}$$

About $\underline{L}'[Q(2,00)]$, we see $Q(2,00)$ consists of the tuples $\boldsymbol{q}$ which satisfy

$$\begin{cases} (q_{00} - q_{11}) + (q_{01} - q_{10}) = \delta_1', \\ (q_{00} - q_{11}) - (q_{01} - q_{10}) = \delta_2', \\ 0 \le q_{\boldsymbol{\sigma}} \le n_{\boldsymbol{\sigma}}, \text{ for all } \boldsymbol{\sigma} \in B^2. \end{cases}$$

Hence, we have

$$\underline{L}'[Q(2,00)] = \sum_{i \in \mathcal{D}_{00}^{((\delta_1'+\delta_2')/2)} \cup \mathcal{D}_{11}^{(-(\delta_1'+\delta_2')/2)} \cup \mathcal{D}_{01}^{((\delta_1'-\delta_2')/2)} \cup \mathcal{D}_{10}^{(-(\delta_1'-\delta_2')/2)}} |r_i|. \tag{4.125}$$

4.6.3 Evaluation of the Improved Optimality Condition for the Case $h = 3$ and $h_0 = 1$

In this subsection, we consider to evaluate the improved optimality condition for the case that there are three reference words and one of them is codeword. Then we need to solve the IPP $\mathcal{P}(Q(3,0))$. We first split the IPP $\mathcal{P}(Q(3,0))$ into 6 sub-IPPs, each of them has at most four variables. Some of these sub-IPPs can be solved directly, the others can be solved by simple iteration.

Without loss of generality, we assume $\delta_2 \ge \delta_3$. Let Ψ_1 denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} + q_{010} + q_{001} + q_{011} = \delta_1', \\ q_{000} - q_{010} + q_{001} - q_{011} \ge \delta_2, \\ q_{000} + q_{010} - q_{001} - q_{011} \ge \delta_3, \\ q_{100} = q_{110} = q_{101} = q_{111} = 0. \end{cases} \tag{4.126}$$

Let Ψ_2 denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} -q_{111} - q_{100} - q_{110} - q_{101} = \delta_1', \\ -q_{111} + q_{100} - q_{110} + q_{101} \ge \delta_2, \\ -q_{111} + q_{100} + q_{110} - q_{101} \ge \delta_3, \\ q_{000} = q_{010} = q_{001} = q_{011} = 0. \end{cases} \tag{4.127}$$

Let Ψ_3 denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} - q_{100} + q_{001} - q_{101} = \delta_1', \\ \delta_2 \le q_{000} + q_{100} + q_{001} + q_{101} \le \delta_2 + 1, \\ q_{000} + q_{100} - q_{001} - q_{101} \ge \delta_3, \\ q_{010} = q_{110} = q_{011} = q_{111} = 0. \end{cases} \tag{4.128}$$

Let Ψ_4 denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} - q_{100} - q_{110} - q_{101} = \delta_1', \\ \delta_2 \le q_{000} + q_{100} - q_{110} + q_{101} \le \delta_2 + 1, \\ \delta_3 \le q_{000} + q_{100} + q_{110} - q_{101} \le \delta_3 + 1, \\ q_{010} = q_{001} = q_{011} = q_{111} = 0. \end{cases} \tag{4.129}$$

Let Ψ_5 denote the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} - q_{100} + q_{010} + q_{001} = \delta_1', \\ \delta_2 \le q_{000} + q_{100} - q_{010} + q_{001} \le \delta_2 + 1, \\ \delta_3 \le q_{000} + q_{100} + q_{010} - q_{001} \le \delta_3 + 1, \\ q_{110} = q_{101} = q_{011} = q_{111} = 0. \end{cases} \tag{4.130}$$

Let Ψ_6 denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} - q_{100} = \delta_1', \ q_{000} + q_{100} \ge \delta_2, \\ q_{010} = q_{001} = q_{110} = q_{011} = q_{101} = q_{111} = 0. \end{cases} \tag{4.131}$$

Lemma 4.14: If $\delta_2 \ge \delta_3$, then we have

$$\underline{L}'[Q(3,0)] = \min_{1 \le j \le 6} \underline{L}'[\Psi_j]. \tag{4.132}$$

Proof: Suppose $\boldsymbol{q}^*$ is a minimal $Q(3,0)$-optimum. Below we prove $\boldsymbol{q}^* \in \cup_{1 \le j \le 6} \Psi_j$ and thus (4.132) holds.

If $q^*_{011} > 0$, then we can see easily that $q^*_{\boldsymbol{\sigma}} = 0$ for all $\boldsymbol{\sigma} \in B^3$ with $\sigma_1 = 1$, and thus $\boldsymbol{q}^* \in \Psi_1$.

If $q^*_{111} > 0$, then we can see easily that $q^*_{\boldsymbol{\sigma}} = 0$ for all $\boldsymbol{\sigma} \in B^3$ with $\sigma_1 = 0$, and thus $\boldsymbol{q}^* \in \Psi_2$.

Below we assume that $q^*_{011} = q^*_{111} = 0$. Clearly, we have

$$q^*_{010} q^*_{101} = 0, \ q^*_{001} q^*_{110} = 0. \tag{4.133}$$

Now we prove that

$$q^*_{110}q^*_{010} = 0. \tag{4.134}$$

Assume in the contrast that $q^*_{110} > 0$ and $q^*_{010} > 0$. By (4.133), we see $q^*_{001} = q^*_{101} = 0$. If

$$q^*_{000} + q^*_{100} + q^*_{010} + q^*_{110} \geq \delta_3 + 2,$$

then the tuple $\boldsymbol{q}$ with

$$\begin{aligned} q_{010} = q^*_{010} - 1,\ q_{110} = q^*_{110} - 1, \\ q_{\boldsymbol{\sigma}} = q^*_{\boldsymbol{\sigma}} \text{ for } \boldsymbol{\sigma} \in B^3 \setminus \{010, 110\}. \end{aligned}$$

is also in $Q(3,0)$, which contradicts the assumption that $\boldsymbol{q}^*$ is minimal. Hence we have

$$\delta_3 \leq q^*_{000} + q^*_{100} + q^*_{010} + q^*_{110} \leq \delta_3 + 1, \tag{4.135}$$

and thus

$$q^*_{000} + q^*_{100} - q^*_{010} - q^*_{110} \leq \delta_3 - 3 < \delta_2,$$

this contradicts that $\boldsymbol{q}^* \in Q(3,0)$. Hence (4.134) holds.

Case 1: $q^*_{101} > 0$ and $q^*_{001} > 0$.

By (4.133), we see $q^*_{010} = q^*_{001} = 0$. Similar to (4.135) we can prove that

$$\delta_2 \leq q^*_{000} + q^*_{100} + q^*_{001} + q^*_{101} \leq \delta_2 + 1,$$

and thus $\boldsymbol{q}^* \in \Psi_3$.

Case 2: $q^*_{110} > 0$ and $q^*_{101} > 0$.

By (4.133), we see $q^*_{001} = q^*_{010} = 0$. If $q^*_{000} = 0$, then $\boldsymbol{q}^* \in \Psi_2$. Now we assume that $q^*_{000} > 0$. Similar to (4.135) we can prove that

$$\begin{aligned} \delta_2 \leq q^*_{000} + q^*_{100} - q^*_{110} + q^*_{101} \leq \delta_2 + 1, \\ \delta_3 \leq q^*_{000} + q^*_{100} + q^*_{110} - q^*_{101} \leq \delta_3 + 1, \end{aligned}$$

and thus $\boldsymbol{q}^* \in \Psi_4$.

Case 3: $q^*_{010} > 0$ and $q^*_{001} > 0$.

By (4.133), we see $q^*_{101} = q^*_{110} = 0$.

If $q^*_{100} = 0$, then $\boldsymbol{q}^* \in \Psi_1$. Now we assume that $q^*_{100} > 0$. Similar to (4.135) we can prove that

$$\delta_2 \leq q^*_{000} + q^*_{100} - q^*_{010} + q^*_{001} \leq \delta_2 + 1,$$
$$\delta_2 \leq q^*_{000} + q^*_{100} + q^*_{010} - q^*_{001} \leq \delta_2 + 1,$$

and thus $\boldsymbol{q}^* \in \Psi_5$.

Case 4: $q^*_{010} > 0$ and $q^*_{001} = q^*_{110} = 0$.

By (4.133), we see $q^*_{101} = 0$. If $q^*_{100} > 0$, similar to (4.135) we can prove that

$$\delta_3 \leq q^*_{000} + q^*_{100} + q^*_{010} \leq \delta_3 + 1,$$

and thus

$$\delta_3 \leq q^*_{000} + q^*_{100} - q^*_{010} \leq \delta_3 - 1 < \delta_2,$$

this contradicts $\boldsymbol{q}^* \in Q(3,0)$. Hence $q^*_{100} = 0$, and then $\boldsymbol{q}^* \in \Psi_1$.

Case 5: $q^*_{001} > 0$ and $q^*_{010} = q^*_{101} = 0$.

By (4.133), we see $q^*_{110} = 0$. If $q^*_{100} = 0$, and $\boldsymbol{q}^* \in \Psi_1$. Now we assume that $q^*_{100} > 0$. Similar to (4.135) we can prove that

$$\delta_2 \leq q^*_{000} + q^*_{100} + q^*_{001} \leq \delta_2 + 1,$$

and thus $\boldsymbol{q}^* \in \Psi_3$.

Case 6: $q^*_{101} > 0$ and $q^*_{010} = q^*_{101} = 0$.

By (4.133), we see $q^*_{010} = 0$. If $q^*_{000} = 0$, and $\boldsymbol{q}^* \in \Psi_2$. Now we assume that $q^*_{000} > 0$. Similar to (4.135) we can prove that

$$\delta_2 \leq q^*_{000} + q^*_{100} + q^*_{101} \leq \delta_2 + 1,$$

and thus $\boldsymbol{q}^* \in \Psi_3$.

Case 7: $q^*_{110} > 0$ and $q^*_{010} = q^*_{101} = 0$.

By (4.133), we see $q^*_{001} = 0$. If $q^*_{000} > 0$, similar to (4.135) we can prove that

$$\delta_3 \leq q^*_{000} + q^*_{100} + q^*_{110} \leq \delta_3 + 1,$$

and thus

$$\delta_3 \leq q^*_{000} + q^*_{100} - q^*_{110} \leq \delta_3 - 1 < \delta_2,$$

this contradicts $\boldsymbol{q}^* \in Q(3,0)$. Hence $q^*_{000} = 0$, and then $\boldsymbol{q}^* \in \Psi_2$.

Case 8: $q^*_{010} = q^*_{001} = q^*_{110} = q^*_{101} = 0$.

We see simply that $\boldsymbol{q}^* \in \Psi_6$. □

Since Ψ_1 and Ψ_2 are sets of the type of $Q_3(\Theta_3)$ and Ψ_4 and Ψ_5 are sets of the type of $Q_3(A_1)$, respectively, the IPPs $\mathcal{P}(\Psi_i)$ can be solved by the methods used in Section 4.4 for $i \in \{1, 2, 4, 5\}$.

To solve the IPP $\mathcal{P}(\Psi_3)$, for $k \geq 0$, let $\Psi_3(k)$ denote the set of the 2^h-tuples $\boldsymbol{q} \in Q_3$ which satisfy (4.2) and

$$\begin{cases} q_{000} + q_{001} = \lfloor (\delta_2 + \delta_1')/2 \rfloor, \\ q_{100} + q_{101} = \lfloor (\delta_2 - \delta_1')/2 \rfloor, \\ q_{000} + q_{100} \geq k, \\ q_{010} = q_{110} = q_{011} = q_{111} = 0. \end{cases} \tag{4.136}$$

Clearly, $\Psi_3 = \Psi_3(\xi)$, where

$$\xi \triangleq \lceil (\lfloor (\delta_2 + \delta_1')/2 \rfloor + \lfloor (\delta_2 - \delta_1')/2 \rfloor + \delta_3)/2 \rceil. \tag{4.137}$$

It is very easy to give a tuple $\boldsymbol{q}_0$ in $\Psi_3(0)$ such that $L'(\boldsymbol{q}_0) = \underline{L}'[\Psi_3(0)]$. And by simple iteration, we can get a tuple $\boldsymbol{q}_\xi$ in $\Psi_3 = \Psi_3(\xi)$ such that $L'(\boldsymbol{q}_\xi) = \underline{L}'[\Psi_3]$ if $\Psi_3 \neq \phi$.

For the IPP $\mathcal{P}(\Psi_6)$, we see easily that

$$\underline{L}'[\Psi_6] = \sum_{i \in \mathcal{D}_{000}^{(\max\{\delta_1', \lceil (\delta_1' + \delta_2)/2 \rceil\})} \cup \mathcal{D}_{100}^{(\lceil (\delta_1' + \delta_2)/2 \rceil - \delta_1')}} |r_i|. \tag{4.138}$$

The number of operations of additions and comparisons of real numbers for solving the IPP $\mathcal{P}(Q(3,0))$ by use of the above methods is of order N.

Chapter 5

Search Centers of the Multistage Chase-Type Decoding

For a positive integer h, a **multistage Chase-Type decoding** [13] is defined as an IDA consisting of successive stages Chase$(\boldsymbol{v}^{(j)}, t_0, \tau_0)$ with $1 \leq j \leq h$, where $\boldsymbol{v}^{(j)}$, called the search center at the j-th stage, is chosen as the hard-decision tuple $\boldsymbol{z}$ for $j = 1$, and the word in V^N which satisfies

$$L(\boldsymbol{v}^{(j)}) = \underline{L}[\cap_{i=1}^{j-1}(V^N \setminus R_{\text{Chase}(\boldsymbol{v}^{(i)},t_0,\tau_0)})] \tag{5.1}$$

for $1 < j \leq h$, where

$$R_{\text{Chase}(\boldsymbol{v}^{(i)},t_0,\tau_0)} \triangleq \{\boldsymbol{x} \in V^N : d_{\text{H},\tau_0+1,N}(\boldsymbol{x}, \boldsymbol{v}^{(i)}) \leq t_0\}$$

is the search region of Chase$(\boldsymbol{v}^{(i)}, t_0, \tau_0)$. Then, (5.1) can be rewritten as

$$= \underline{L}[\{\boldsymbol{x} \in V^N : d_{\text{H},\tau_0+1,N}(\boldsymbol{x}, \boldsymbol{v}^{(i)}) \geq t_0 + 1, \text{ for } 1 \leq i < j\}]$$

We consider to determine the first a few search centers for this algorithm in this chapter.

Clearly, $p_{1,\tau_0}(\boldsymbol{v}^{(j)}) = p_{1,\tau_0}(\boldsymbol{z})$ for any j. Let $\boldsymbol{r}' \triangleq p_{\tau_0+1,N}(\boldsymbol{r})$ and $\boldsymbol{z}' \triangleq p_{\tau_0+1,N}(\boldsymbol{z})$. For $\boldsymbol{u} \in V^n$ with $n \triangleq N - \tau_0$, let

$$L^*(\boldsymbol{u}) \triangleq \sum_{1 \leq i \leq n, u_i \neq z'_i} |r'_i|. \tag{5.2}$$

Then the problem for determining the search centers $\boldsymbol{v}^{(j)}$ is equivalent to the problem for determining a series of n-tuples $\boldsymbol{v}_j \in V^n$ such that $\boldsymbol{v}_1 = \boldsymbol{z}'$, and for $j > 1$, $\boldsymbol{v}_j$ satisfies

$$\boldsymbol{v}_j \in V^n_{\tau,\tau,\ldots,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j-1}), \tag{5.3}$$

$$L^*(\boldsymbol{v}_j) = \min_{\boldsymbol{u} \in V^n_{\tau,\tau,\ldots,\tau}(\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_{j-1})} L^*(\boldsymbol{u}), \tag{5.4}$$

where $\tau \triangleq t_0 + 1$.

It is known that

$$\boldsymbol{z}' + \boldsymbol{v}_1 = (\underbrace{0, \ldots, 0}_{n}), \tag{5.5}$$

$$\boldsymbol{z}' + \boldsymbol{v}_2 = (\underbrace{1, \ldots, 1}_{\tau}, \underbrace{0, \ldots, 0}_{n-\tau}), \tag{5.6}$$

$$\boldsymbol{z}' + \boldsymbol{v}_3 = (\underbrace{1, \ldots, 1}_{\lfloor \tau/2 \rfloor}, \underbrace{0, \ldots, 0}_{\lceil \tau/2 \rceil}, \underbrace{1, \ldots, 1}_{\lceil \tau/2 \rceil}, \underbrace{0\ , \ \ldots\ , \ 0}_{n-\tau-\lceil \tau/2 \rceil}). \tag{5.7}$$

The n-tuples $\boldsymbol{v}_4$ and $\boldsymbol{v}_5$ are determined in Sections 5.1 and 5.2 respectively. The computational complexity of the determination is shown to be of order τ.

5.1 Determination of the Search Center of the Fourth Stage

Assume $n^* \triangleq n - \tau - \lceil \tau/2 \rceil \geq 0$. From (2.11) and (5.3) to (5.7), the third search center must be an n-tuple $\boldsymbol{u} \in V^n_{\tau,\tau,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ for which the n-tuple $\boldsymbol{z}' + \boldsymbol{u}$ has the form of

$$(\underbrace{1, \ldots, 1}_{x_4}, \underbrace{0, \ldots, 0}_{\lfloor \tau/2 \rfloor - x_4}, \underbrace{1, \ldots, 1}_{x_3}, \underbrace{0, \ldots, 0}_{\lceil \tau/2 \rceil - x_3}, \underbrace{1, \ldots, 1}_{x_2}, \underbrace{0, \ldots, 0}_{\lceil \tau/2 \rceil - x_2}, \underbrace{1, \ldots, 1}_{x_1}, \underbrace{0, \ldots, 0}_{n^* - x_1}), \tag{5.8}$$

where x_1, x_2, x_3 and x_4 satisfy

$$\begin{cases} 0 \leq x_1 \leq n^*, \ 0 \leq x_2 \leq \lceil \tau/2 \rceil, \\ 0 \leq x_3 \leq \lceil \tau/2 \rceil, \ 0 \leq x_4 \leq \lfloor \tau/2 \rfloor, \end{cases} \tag{5.9}$$

and

$$\begin{cases} w_{\mathrm{H}}(\boldsymbol{u} + \boldsymbol{v}_1) = x_4 + x_3 + x_2 + x_1 \geq \tau, \\ w_{\mathrm{H}}(\boldsymbol{u} + \boldsymbol{v}_2) = (\tau - x_4 - x_3) + x_2 + x_1 \geq \tau, \\ w_{\mathrm{H}}(\boldsymbol{u} + \boldsymbol{v}_3) = (\lfloor \tau/2 \rfloor - x_4) + x_3 + (\lceil \tau/2 \rceil - x_2) + x_1 \geq \tau, \end{cases} \tag{5.10}$$

i.e.

$$\begin{cases} x_1+x_2+x_3+x_4 \geq \tau, \\ x_1+x_2-x_3-x_4 \geq 0, \\ x_1-x_2+x_3-x_4 \geq 0. \end{cases} \tag{5.11}$$

Let Ω_4 denote the set of pairs $\pi \triangleq (x_1, x_2, x_3, x_4)$ which satisfy (5.9) and (5.11). For $\pi = (x_1, x_2, x_3, x_4) \in \Omega_4$, let $\boldsymbol{u}(\pi)$ denote the n-tuple of V^n for which $\boldsymbol{z}' + \boldsymbol{u}(\pi)$ is given by (5.8). Let π^* be a pair of Ω_4 which satisfies $\boldsymbol{v}_4 = \boldsymbol{u}(\pi^*)$. For any two pairs π_1 and π_2 of Ω_4, we say $\pi_1 \prec \pi_2$ if $L^*(\boldsymbol{u}(\pi_1)) < L^*(\boldsymbol{u}(\pi_2))$. Clearly, there is no pair π such that $\pi \prec \pi^*$. Below we consider to determine the pair $\pi^* = (x_1^*, x_2^*, x_3^*, x_4^*)$.

Lemma 5.1: About the pair π^*, we have the following five simple results:
1) $x_2^* = x_3^*$;
2) $x_4^* \leq x_1^* \leq x_4^* + 1$;
3) $\Delta \triangleq x_1^* + x_2^* + x_3^* + x_4^* - \tau \leq 1$;
4) $\Delta^* \triangleq \Delta + (x_1^* - x_4^*) \leq 1$;
5) $\Delta^* = \lceil \tau/2 \rceil - \lfloor \tau/2 \rfloor$.

Proof: 1) One hand, if $x_2^* > x_3^*$, then we have $(x_1^*, x_3^*, x_2^*, x_4^*) \prec \pi^*$, contradictory. On the other hand, if $x_2^* < x_3^*$, then from $x_1^* + x_2^* - x_3^* - x_4^* \geq 0$ we see $x_1^* - 1 \geq x_4^*$ and then $(x_1^* - 1, x_2^* + 1, x_3^*, x_4^*) \prec \pi^*$, contradictory.

2) From 1) and (5.11), we see that $x_4^* \leq x_1^*$. If $x_1^* \geq x_4^* + 2$ and $x_2^* < \lceil \tau/2 \rceil$, then we have $(x_1^* - 2, x_2^* + 1, x_3^* + 1, x_4^*) \prec \pi^*$, contradictory. If $x_1^* \geq x_4^* + 2$ and $x_2^* = \lceil \tau/2 \rceil$, then we have $(0, x_2^*, x_3^*, 0) \prec \pi^*$, contradictory. Hence we must have $x_1^* \leq x_4^* + 1$.

3) If $\Delta \geq 2$, then from 1) and 2) we see easily that $x_1^* \cdot x_4^* \neq 0$ or $x_2^* \cdot x_3^* \neq 0$, and thus we have $(x_1^* - 1, x_2^*, x_3^*, x_4^* - 1) \prec \pi^*$ or $(x_1^*, x_2^* - 1, x_3^* - 1, x_4^*) \prec \pi^*$, contradictory.

4) If $\Delta^* \geq 2$, then by 2) and 3) we see $\Delta = x_1^* - x_4^* = 1$ and thus $(x_1^* - 1, x_2^*, x_3^*, x_4^*) \prec \pi^*$, contradictory.

5) $\Delta^* = \lceil \tau/2 \rceil - \lfloor \tau/2 \rfloor$ follows from $0 \leq \Delta^* \leq 1$ and $\tau + \Delta^* = 2(x_1^* + x_2^*)$ is even. □

Let T denote the set of pairs π with the form of $(\lceil\tau/2\rceil - x, x, x, \lfloor\tau/2\rfloor - x)$, where integer x satisfies

$$x_0 \triangleq \max\{0, \lceil\tau/2\rceil - n^*\} \leq x \leq \lfloor\tau/2\rfloor. \tag{5.12}$$

Theorem 5.1: There is a pair π^* with

$$\pi^* \in \begin{cases} T, & \text{if } \tau \text{ is even,} \\ T \cup \{(0, \lceil\tau/2\rceil, \lceil\tau/2\rceil, 0)\}, & \text{otherwise,} \end{cases} \tag{5.13}$$

such that $\boldsymbol{v}_4 = \boldsymbol{u}(\pi^*)$.

Proof: If $\Delta = 0$, from Lemma 5.1, we see easily that

$$x_2^* = x_3^*,\ x_1^* - x_4^* = \lceil\tau/2\rceil - \lfloor\tau/2\rfloor,\ x_2^* + x_4^* = \lfloor\tau/2\rfloor, \tag{5.14}$$

and then from $0 \leq x_1^* \leq n^*$ we see easily $\pi^* \in T$.

If $\Delta = 1$, from Lemma 5.1, we see that

$$\lceil\tau/2\rceil - \lfloor\tau/2\rfloor = \Delta^* \geq \Delta = 1, \tag{5.15}$$

this means that $\Delta^* = 1$ and τ is odd. Thus by Lemma 5.1 we have

$$x_2^* = x_3^*,\ x_1^* - x_4^* = \Delta^* - \Delta = 0,\ x_2^* + x_4^* = \lceil\tau/2\rceil. \tag{5.16}$$

Furthermore, we conclude that x_4^* must be zero, if in the contrast $x_4^* > 0$, then we have $(x_1^*, x_2^*, x_3^*, x_4^* - 1) \prec \pi^*$, contradictory. Hence we have π^* must be the pair $(0, \lceil\tau/2\rceil, \lceil\tau/2\rceil, 0)$ if $\Delta = 1$. □

Assume $\boldsymbol{v}_4 = \boldsymbol{u}(\lceil\tau/2\rceil - x^*, x^*, x^*, \lfloor\tau/2\rfloor - x^*)$. Write

$$\eta(x) \triangleq -|r'_{\tau+2\lceil\tau/2\rceil-x}| + |r'_{\tau+x+1}| + |r'_{\lfloor\tau/2\rfloor+x+1}| - |r'_{\lfloor\tau/2\rfloor-x}|, \tag{5.17}$$

and $\pi_0 \triangleq (\lceil\tau/2\rceil - x_0, x_0, x_0, \lfloor\tau/2\rfloor - x_0)$, where x_0 is defined in (5.12). Then

$$L^*(\boldsymbol{v}_3) - L^*(\boldsymbol{u}(\pi_0)) = \sum_{x_0 \leq x < x^*} \eta(x). \tag{5.18}$$

Since $\eta(x)$ is monotonically decreasing, we can find the integer x^* easily. Indeed, we can determine the integer x^* with at most $4\lfloor\tau/2\rfloor \leq 2\tau$ operations of additions and comparisons of real numbers. Hence, the number of operations of additions and comparisons of real numbers for determining $\boldsymbol{v}_4$ is of order τ.

5.2 Determination of the Search Center of the Fifth Stage

5.2.1 Main Algorithm for Determing v_5

According to Theorem 5.1, the determination of the fifth research center $\boldsymbol{v}_5$ is divided into two cases:

Case A: τ is odd and $\boldsymbol{v}_4 = \boldsymbol{u}(0, \lceil\tau/2\rceil, \lceil\tau/2\rceil, 0)$.

Let T' denote the set of pairs π with the form of $(\lceil\tau/2\rceil - x, x, x, \lfloor\tau/2\rfloor - x)$, where integer x satisfies $x_0 \le x \le \lfloor\lceil\tau/2\rceil/2\rfloor$. T' is a subset of T.

Theorem 5.2: Assume τ is odd and $\boldsymbol{v}_3 = \boldsymbol{u}(0, \lceil\tau/2\rceil, \lceil\tau/2\rceil, 0)$.
1. If $n^* < \lceil\lceil\tau/2\rceil/2\rceil$, then $V^n_{\tau,\tau,\tau,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) = \phi$.
2. If $n^* \ge \lceil\lceil\tau/2\rceil/2\rceil$, then $\boldsymbol{v}_5 = \boldsymbol{u}(\acute{\pi})$ for some pair $\acute{\pi} \in T'$.

Proof: We only consider the n-tuples $\boldsymbol{u} \in V^n_{\tau,\tau,\tau,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4)$ for which the n-tuples $\boldsymbol{z}' + \boldsymbol{u}$ have the form of (5.8) with integers x_1, x_2, x_3, x_4 satisfying (5.9) and (5.11) and $w(\boldsymbol{u} + \boldsymbol{v}_4) = x_4 + (\lceil\tau/2\rceil - x_3) + (\lceil\tau/2\rceil - x_2) + x_1 \ge \tau$, i.e.

$$x_1 - x_2 - x_3 + x_4 \ge \tau - 2\lceil\tau/2\rceil = -1. \tag{5.19}$$

Let Ω^* denote the set of pairs (x_1, x_2, x_3, x_4) which satisfy (5.9) and (5.11) and (5.19), it is a subset of Ω_4. If $(x_1, x_2, x_3, x_4) \in \Omega^*$, then we see easily $x_1 \ge \max\{x_2, x_3, x_4\}$ and thus $4x_1 \ge \tau$, i.e. $x_1 \ge \lceil\lceil\tau/2\rceil/2\rceil$. Thus by $x_1 \le n^*$ we see $V^n_{\tau,\tau,\tau,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4) = \phi$ if $n^* < \lceil\lceil\tau/2\rceil/2\rceil$. Now we assume $n^* \ge \lceil\lceil\tau/2\rceil/2\rceil$.

Let $\acute{\pi} = (\acute{x}_1, \acute{x}_2, \acute{x}_3, \acute{x}_4)$ denote the pair of Ω^* which satisfies $\boldsymbol{v}_5 = \boldsymbol{u}(\acute{\pi})$. For any two pairs π_1 and π_2 of Ω^*, we say $\pi_1 \prec^* \pi_2$ if $L^*(\boldsymbol{u}(\pi_1)) < L^*(\boldsymbol{u}(\pi_2))$. Clearly, $\acute{x}_2 \le \acute{x}_3$, $\acute{x}_1 \ge \max\{\acute{x}_3, \acute{x}_4\}$ and there is no pair π such that $\pi \prec^* \acute{\pi}$.

At first, we consider the case of $\acute{x}_1 - \acute{x}_2 - \acute{x}_3 + \acute{x}_4 \ge 0$. Clearly, $\acute{x}_2 \le \acute{x}_3 \le \acute{x}_4$ and $\acute{x}_1 + \acute{x}_2 \ge \acute{x}_3 + \acute{x}_4$. If $\acute{x}_3 > \acute{x}_2$, then $\acute{x}_1 > \acute{x}_4$ and $(\acute{x}_1 - 1, \acute{x}_2 + 1, \acute{x}_3, \acute{x}_4) \prec^* \acute{\pi}$, contradictory. Hence, $\acute{x}_2 = \acute{x}_3$. If $\acute{x}_1 \ge \acute{x}_4 + 2$, then $(\acute{x}_1 - 1, \acute{x}_2, \acute{x}_3 + 1, \acute{x}_4) \prec^* \acute{\pi}$, contradictory. Hence, $\acute{x}_1 - 1 \le \acute{x}_4$. If $\acute{x}_1 = \acute{x}_4$, with respect to that τ is odd we have $\acute{x}_1 + \acute{x}_2 + \acute{x}_3 + \acute{x}_4 > \tau$ and thus by $\acute{x}_4 = \acute{x}_1 \ge \lceil\lceil\tau/2\rceil/2\rceil \ge 1$ we see $(\acute{x}_1, \acute{x}_2, \acute{x}_3, \acute{x}_4 - 1) \prec^* \acute{\pi}$, contradictory. Hence we have $\acute{x}_1 = \acute{x}_4 + 1$. If

$\acute{x}_1+\acute{x}_2+\acute{x}_3+\acute{x}_4 > \tau$, then $(\acute{x}_1-1, \acute{x}_2, \acute{x}_3, \acute{x}_4) \prec^* \acute{\pi}$, contradictory. Hence we have $\acute{x}_1+\acute{x}_2+\acute{x}_3+\acute{x}_4 = \tau$ and

$$\acute{x}_1 - 1 = \acute{x}_4 \geq \acute{x}_2 = \acute{x}_3, \ \acute{x}_2 + \acute{x}_4 = \lceil \tau/2 \rceil - 1. \tag{5.20}$$

By (5.20) and $\acute{x}_1 \leq n^*$, we see

$$x_0 \leq \acute{x}_2 \leq \lfloor (\lceil \tau/2 \rceil - 1)/2 \rfloor = \begin{cases} \lfloor \lceil \tau/2 \rceil /2 \rfloor, & \text{if } \lceil \tau/2 \rceil \text{ is odd,} \\ \lfloor \lceil \tau/2 \rceil /2 \rfloor - 1, & \text{if } \lceil \tau/2 \rceil \text{ is even,} \end{cases} \tag{5.21}$$

and thus $\acute{\pi} \in T'$.

Now we consider the case of $\acute{x}_1 - \acute{x}_2 - \acute{x}_3 + \acute{x}_4 = -1$, i.e. $\acute{x}_2+\acute{x}_3 = \acute{x}_1+\acute{x}_4+1$. we see easily that $\acute{x}_2 + \acute{x}_3 \geq \lceil \tau/2 \rceil$ and $\acute{x}_4 < \min\{\acute{x}_2, \tau^*\}$. If $\acute{x}_1 > \acute{x}_3$, then $(\acute{x}_1 - 1, \acute{x}_2, \acute{x}_3, \acute{x}_4 + 1) \prec^* \acute{\pi}$, contradictory. Hence $\acute{x}_1 = \acute{x}_3$ and $\acute{x}_2 = \acute{x}_4 + 1$. If $\acute{x}_2 + \acute{x}_3 > \lceil \tau/2 \rceil$, then $\acute{x}_3 > 0$ and with respect to τ is odd we see $(\acute{x}_1, \acute{x}_2 - 1, \acute{x}_3 - 1, \acute{x}_4) \prec^* \acute{\pi}$, contradictory. Hence $\acute{x}_2 + \acute{x}_3 = \lceil \tau/2 \rceil$. If $\acute{x}_1 \geq \acute{x}_2 + 2$, then $(\acute{x}_1 - 1, \acute{x}_2 + 1, \acute{x}_3 - 1, \acute{x}_4 + 1) \prec^* \acute{\pi}$, contradictory. Hence $\acute{x}_1 \leq \acute{x}_2 + 1$ and thus we have

$$\acute{x}_4 + 1 = \acute{x}_2 \leq \acute{x}_1 = \acute{x}_3 \leq \acute{x}_2 + 1, \ \acute{x}_2 + \acute{x}_3 = \lceil \tau/2 \rceil, \tag{5.22}$$

and sequently,

$$\acute{x}_1 = \lceil \lceil \tau/2 \rceil /2 \rceil, \ \acute{x}_2 = \lfloor \lceil \tau/2 \rceil /2 \rfloor, \ \acute{x}_3 = \lceil \lceil \tau/2 \rceil /2 \rceil, \ \acute{x}_4 = \lfloor \lceil \tau/2 \rceil /2 \rfloor - 1. \tag{5.23}$$

If $\lceil \tau/2 \rceil$ is odd, then $\lceil \lceil \tau/2 \rceil /2 \rceil = \lfloor \lceil \tau/2 \rceil /2 \rfloor + 1$ and thus $(\acute{x}_1 - 1, \acute{x}_2 + 1, \acute{x}_3 - 1, \acute{x}_4 + 1) \prec^* \acute{\pi}$, contradictory. Hence $\lceil \tau/2 \rceil$ must be even, i.e. $\lceil \lceil \tau/2 \rceil /2 \rceil = \lfloor \lceil \tau/2 \rceil /2 \rfloor$, and thus

$$\acute{\pi} = (\lceil \tau/2 \rceil - \lfloor \lceil \tau/2 \rceil /2 \rfloor, \lfloor \lceil \tau/2 \rceil /2 \rfloor, \lfloor \lceil \tau/2 \rceil /2 \rfloor, \lfloor \tau/2 \rfloor - \lfloor \lceil \tau/2 \rceil /2 \rfloor), \tag{5.24}$$

it is a pair of T'. □

Clearly, the number of operations of additions and comparisons of real numbers for determining $\boldsymbol{v}_5$ is of order τ if τ is odd and $\boldsymbol{v}_4 = \boldsymbol{u}(0, \lceil \tau/2 \rceil, \lceil \tau/2 \rceil, 0)$.

Case B: There is an integer x^* with $x_0 \leq x^* \leq \lfloor \tau/2 \rfloor$ such that

$$\boldsymbol{v}_4 = \boldsymbol{u}(\lceil \tau/2 \rceil - x^*, x^*, x^*, \lfloor \tau/2 \rfloor - x^*). \tag{5.25}$$

Clearly, we have $n \geq 3\lceil\tau/2\rceil$. We only consider the n-tuples $\boldsymbol{u}$ of $V^n_{\tau,\tau,\tau,\tau}(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4)$ for which $\boldsymbol{z}' + \boldsymbol{u}$ have the form of

$$\begin{aligned}(&\underbrace{1,\ldots,1}_{y_8},\underbrace{0,\ldots,0}_{\lfloor\tau/2\rfloor-x^*-y_8},\underbrace{1,\ldots,1}_{y_7},\underbrace{0,\ldots,0}_{x^*-y_7},\underbrace{1,\ldots,1}_{y_6},\underbrace{0,\ldots,0}_{x^*-y_6},\\ &\underbrace{1,\ldots,1}_{y_5},\underbrace{0,\ldots,0}_{\lceil\tau/2\rceil-x^*-y_5},\underbrace{1,\ldots,1}_{y_4},\underbrace{0,\ldots,0}_{x^*-y_4},\underbrace{1,\ldots,1}_{y_3},\underbrace{0,\ldots,0}_{\lceil\tau/2\rceil-x^*-y_3},\\ &\underbrace{1,\ldots,1}_{y_2},\underbrace{0,\ldots,0}_{\lceil\tau/2\rceil-x^*-y_2},\underbrace{1,\ldots,1}_{y_1},\underbrace{0,\ldots,0}_{n^*+x^*-\lceil\tau/2\rceil-y_1}),\end{aligned} \tag{5.26}$$

where integers $y_1, y_2, \ldots, y_8$ satisfy

$$\begin{cases} 0 \leq y_1 \leq n^* + x^* - \lceil\tau/2\rceil, \ 0 \leq y_8 \leq \lfloor\tau/2\rfloor - x^*, \\ 0 \leq y_2, \ y_3, \ y_5 \leq \lceil\tau/2\rceil - x^*, \ 0 \leq y_4, \ y_6, \ y_7 \leq x^*, \end{cases} \tag{5.27}$$

and

$$\begin{cases} w(\boldsymbol{u}+\boldsymbol{v}_1) = y_8 + y_7 + y_6 + y_5 + y_4 + y_3 + y_2 + y_1 \geq \tau, \\ w(\boldsymbol{u}+\boldsymbol{v}_2) = (\tau - y_8 - y_7 - y_6 - y_5) + y_4 + y_3 + y_2 + y_1 \geq \tau, \\ w(\boldsymbol{u}+\boldsymbol{v}_3) = (\lfloor\tau/2\rfloor - y_8 - y_7) + y_6 + y_5 + \\ \qquad\qquad (\lceil\tau/2\rceil - y_4 - y_3) + y_2 + y_1 \geq \tau, \\ w(\boldsymbol{u}+\boldsymbol{v}_4) = (\lfloor\tau/2\rfloor - x^* - y_8) + y_7 + (x^* - y_6) + y_5 + \\ \qquad\qquad (x^* - y_4) + y_3 + (\lceil\tau/2\rceil - x^* - y_2) + y_1 \geq \tau, \end{cases} \tag{5.28}$$

i.e.

$$\begin{cases} y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 \geq \tau, \\ y_1 + y_2 + y_3 + y_4 - y_5 - y_6 - y_7 - y_8 \geq 0, \\ y_1 + y_2 - y_3 - y_4 + y_5 + y_6 - y_7 - y_8 \geq 0, \\ y_1 - y_2 + y_3 - y_4 + y_5 - y_6 + y_7 - y_8 \geq 0. \end{cases} \tag{5.29}$$

Let Ω_5 denote the set of pairs $\boldsymbol{\alpha} \triangleq (y_1, y_2, \ldots, y_8)$ which satisfy (5.27) and (5.29). For $\boldsymbol{\alpha} \in \Omega_5$, let $\boldsymbol{u}(\boldsymbol{\alpha})$ denote the n-tuple of V^n for which $\boldsymbol{z}' + \boldsymbol{u}(\boldsymbol{\alpha})$ is defined by (5.26). If two pairs $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ of Ω_5 satisfy $L^*(\boldsymbol{u}(\boldsymbol{\alpha})) < L^*(\boldsymbol{u}(\boldsymbol{\alpha}'))$, we say $\boldsymbol{\alpha} \prec \boldsymbol{\alpha}'$. Let $\boldsymbol{\alpha}^* \triangleq (y_1^*, y_2^*, \ldots, y_8^*)$ denote the pair which satisfies $\boldsymbol{v}_5 = \boldsymbol{u}(\boldsymbol{\alpha}^*)$. Clearly, $\boldsymbol{\alpha}^* \not\prec \boldsymbol{\alpha}$ for all pair $\boldsymbol{\alpha}$ of Ω_5. We have

Lemma 5.2: 1) $0 \leq \sum_{i=1}^{8} y_i^* - \tau \leq 1$. 2) If $\sum_{i=1}^{8} y_i^* = \tau + 1$, then τ is odd.

Proof: 1) Assume in contrast $\sum_{i=1}^{8} y_i^* - \tau \geq 2$. Clearly, $y_8^* = 0$ and for each pair (i,j) of $\{(2,7),(3,6),(4,5),(4,6),(4,7),(6,7)\}$ we have $y_i^* \cdot y_j^* = 0$. If $y_4^* > 0$, then we have $y_5^* = y_6^* = y_7^* = 0$ and thus $(y_1^*, y_2^*, y_3^*, y_4^* - 1, y_5^*, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Hence we have $y_4^* = 0$. Similarly, we can get $y_6^* = y_7^* = 0$. If $\max\{y_2^*, y_3^*, y_5^*\} > 0$, say y_2^* is the biggest one, then $(y_1^*, y_2^* - 1, y_3^*, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Thus $y_2^* = y_3^* = y_5^* = 0$ and we also have $(y_1^* - 1, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory.

2) Assume in contrast τ is even. Since $\sum_{i=1}^{8} y_i^* = \tau + 1$ is odd, we see

$$\begin{cases} y_1^* + y_2^* + y_3^* + y_4^* - y_5^* - y_6^* - y_7^* - y_8^* \geq 1, \\ y_1^* + y_2^* - y_3^* - y_4^* + y_5^* + y_6^* - y_7^* - y_8^* \geq 1, \\ y_1^* - y_2^* + y_3^* - y_4^* + y_5^* - y_6^* + y_7^* - y_8^* \geq 1. \end{cases}$$

If $y_i^* > 0$ for some i, say $i = 1$, we see $(y_1^* - 1, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. □

Let S_1 denote the set of pairs $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8) \in \Omega_5$ given by

$$\begin{cases} y_1 = y_7 = 0, \ y_2 = \lceil \tau/2 \rceil - k, \ y_8 = \lfloor \tau/2 \rfloor - k, \\ y_3 = y_5 = k - x^*, \ y_4 = y_6 = x^*, \end{cases} \tag{5.30}$$

where integer k satisfies $\max\{x^* + \lfloor (\tau+1)/4 \rfloor + 1, \lceil \tau/2 \rceil - n^*\} \leq k \leq \lfloor \tau/2 \rfloor$.

Let S_2 denote the set of pairs $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8) \in \Omega_5$ given by

$$\begin{cases} y_1 = y_7 = x - k, \ y_4 = y_6 = k, \\ y_3 = y_5 = 0, \ y_2 = \lceil \tau/2 \rceil - x^*, \ y_8 = \lfloor \tau/2 \rfloor - x^*, \end{cases} \tag{5.31}$$

where integer k satisfies $\max\{0, \lceil \tau/2 \rceil - n^*\} \leq k < x^* - \lfloor (\tau+1)/4 \rfloor$.

Let S_3 denote the set of pairs $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8) \in \Omega_5$ which satisfy

$$\begin{cases} y_1 + y_2 = \lceil \tau/2 \rceil - k, \ y_3 + y_4 = y_5 + y_6 = k, \\ y_7 + y_8 = \lfloor \tau/2 \rfloor - k, \ y_1 + y_3 + y_5 + y_7 = \lceil \tau/2 \rceil, \end{cases} \tag{5.32}$$

where $d_1^* \triangleq \max\{0, \lceil \tau/2 \rceil - n^*, x^* - \lfloor (\tau+1)/4 \rfloor, \lceil \tau/2 \rceil - \lfloor n^*/2 \rfloor - x^*\} \leq k \leq d_2^* \triangleq \min\{\lfloor \tau/2 \rfloor, x^* + \lfloor (\tau+1)/4 \rfloor, n^* + \lfloor \tau/2 \rfloor - x^*, \lceil \tau/2 \rceil + \lfloor \tau/4 \rfloor - x^*\}$.

Theorem 5.3: If $\sum_{j=1}^{8} y_j^* = \tau$, then $\boldsymbol{\alpha}^*$ must belong to $S_1 \cup S_2 \cup S_3$.

Proof: Let $d^* \triangleq \lceil \tau/2 \rceil - y_1^* - y_2^*$. By (5.29) and $\sum_{i=1}^{8} y_i^* = \tau$, we see

$$\begin{cases} y_3^* + y_4^* \geq d^*, \ y_5^* + y_6^* \geq d^*, \\ y_1^* + y_3^* + y_5^* + y_7^* \geq \lceil \tau/2 \rceil. \end{cases}$$

By (5.27), we see easily $d^* \geq \lceil \tau/2 \rceil - n^*$. We conclude that $d^* \geq 0$. Assume in contrast $d^* < 0$. Clearly, we have $y_1^* > 0$ and $y_5^* + y_7^* < \lceil \tau/2 \rceil$, i.e. either $y_5^* < \lceil \tau/2 \rceil - x^*$ or $y_7^* < x^*$, and thus we have either $(y_1^* - 1, y_2^*, y_3^*, y_4^*, y_5^* + 1, y_6^*, y_7^*, y_8^*) \prec \alpha^*$ or $(y_1^* - 1, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*, y_7^* + 1, y_8^*) \prec \alpha^*$, contradictory.

If $d^* > \lfloor \tau/2 \rfloor$, then $d^* = \lceil \tau/2 \rceil > \lfloor \tau/2 \rfloor$ and $y_1^* + y_2^* = 0$ and thus by $y_3^* + y_4^* \geq \lceil \tau/2 \rceil$ and $y_5^* + y_6^* \geq \lceil \tau/2 \rceil$ we see $\sum_{j=1}^{8} y_j^* > \tau$, contradictory. Hence, we have $d^* \leq \lfloor \tau/2 \rfloor$.

If $y_3^* + y_4^* > d^*$, we have $y_3^* > 0$ or $y_4^* > 0$ and by $y_5^* + y_7^* \leq \tau - (y_1^* + y_2^* + y_3^* + y_4^*) < \lfloor \tau/2 \rfloor$ we see $y_5^* < \lceil \tau/2 \rceil - x^*$ or $y_7^* < x^*$, say $y_3^* > 0$ and $y_7^* < x^*$, then we have $(y_1^*, y_2^*, y_3^* - 1, y_4^*, y_5^*, y_6^*, y_7^* + 1, y_8^*) \prec \alpha^*$, contradictory. Hence, we have $y_3^* + y_4^* = d^*$.

Furthermore, we can conclude that $y_5^* + y_6^* = d^*$. Assume in contrast $y_5^* + y_6^* > d^*$. By $y_7^* + y_8^* < \lfloor \tau/2 \rfloor - d^*$ we see $y_7^* < x^*$ or $y_8^* < \lfloor \tau/2 \rfloor - x^*$. If $y_6^* > 0$, then $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^* - 1, y_7^* + 1, y_8^*) \prec \alpha^*$ or $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^* - 1, y_7^*, y_8^* + 1) \prec \alpha^*$, contradictory. Hence $y_6^* = 0$ and $y_5^* > d^*$. If $y_7^* < x^*$, we have $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^* - 1, y_6^*, y_7^* + 1, y_8^*) \prec \alpha^*$, contradictory. Hence $y_7^* = x^*$ and $y_8^* < \lfloor \tau/2 \rfloor - x^*$. If $y_1^* + y_3^* + y_5^* + y_7^* > \lceil \tau/2 \rceil$, then $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^* - 1, y_6^*, y_7^*, y_8^* + 1) \prec \alpha^*$, contradictory. Hence $y_1^* + y_3^* + y_5^* + y_7^* = \lceil \tau/2 \rceil$ and thus $y_3^* < \lceil \tau/2 \rceil - y_7^* = \lceil \tau/2 \rceil - x^*$. By $d^* < y_5^* \leq \lceil \tau/2 \rceil - x^*$, we see $y_1^* + y_2^* = \lceil \tau/2 \rceil - d^* > 0$, i.e. $y_1^* > 0$ or $y_2^* > 0$, and thus $(y_1^* - 1, y_2^*, y_3^* + 1, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \alpha^*$ or $(y_1^*, y_2^* - 1, y_3^* + 1, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \alpha^*$, contradictory. Hence $y_5^* + y_6^* = d^*$ and thus

$$\begin{cases} y_1^* + y_2^* = \lceil \tau/2 \rceil - d^*, \ y_3^* + y_4^* = y_5^* + y_6^* = d^*, \\ y_7^* + y_8^* = \lfloor \tau/2 \rfloor - d^*, \ y_1^* + y_3^* + y_5^* + y_7^* \geq \lceil \tau/2 \rceil. \end{cases}$$

Consider the case of $y_1^* + y_3^* + y_5^* + y_7^* > \lceil \tau/2 \rceil$. Clearly, we have

$$\begin{cases} y_1^* = \max\{0, x^* - d^*\}, \ y_2^* = \min\{\lceil \tau/2 \rceil - x^*, \lceil \tau/2 \rceil - d^*\}, \\ y_3^* = y_5^* = \max\{0, d^* - x^*\}, \ y_4^* = y_6^* = \min\{x^*, d^*\}, \\ y_7^* = \max\{0, x^* - d^*\}, \ y_8^* = \min\{\lfloor \tau/2 \rfloor - x^*, \lfloor \tau/2 \rfloor - d^*\}. \end{cases}$$

and $d^* > x^* + \lfloor(\tau+1)/4\rfloor$ or $d^* < x^* - \lfloor(\tau+1)/4\rfloor$, i.e. $\boldsymbol{\alpha}^* \in S_1 \cup S_2$.

Consider the case of $y_1^* + y_3^* + y_5^* + y_7^* = \lceil\tau/2\rceil$. Clearly, we have $x^* - \lfloor(\tau+1)/4\rfloor \le d^* \le x^* + \lfloor(\tau+1)/4\rfloor$. Let $M^*(k) \triangleq \min\{n^* + x^* - \lceil\tau/2\rceil, \lceil\tau/2\rceil - k\} + 2\min\{\lceil\tau/2\rceil - x^*, k\} + \min\{\lfloor\tau/2\rfloor - k, x^*\}$. We can show easily that $M^*(k) \ge \lceil\tau/2\rceil$ if and only if k satisfies

$$\lceil\tau/2\rceil - \lfloor n^*/2\rfloor \le k + x^* \le \min\{n^* + \lfloor\tau/2\rfloor, \lceil\tau/2\rceil + \lfloor\tau/4\rfloor\}. \tag{5.33}$$

Then we see easily $d_1^* \le d^* \le d_2^*$, i.e. $\boldsymbol{\alpha}^* \in S_3$. □

Let S_4 denote the set of pairs $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8) \in \Omega_5$ which satisfy

$$\begin{cases} y_1 + y_2 = \lceil\tau/2\rceil - k, \ y_3 + y_4 = y_5 + y_6 = k, \\ y_7 = \lceil\tau/2\rceil - k, \ y_8 = 0, \ y_1 + y_3 + y_5 + y_7 = \lceil\tau/2\rceil, \end{cases} \tag{5.34}$$

where integer k satisfies $d_1' \triangleq \max\{\lceil\tau/2\rceil - n^*, \lceil\tau/2\rceil - x^*, \lceil x^*/2\rceil\} \le k \le d_2' \triangleq \min\{\lceil\tau/2\rceil, 2x^*, n^* + \lceil\tau/2\rceil - x^*, \lceil\tau/2\rceil + \lfloor\lceil\tau/2\rceil/2\rfloor - x^*\}$.

Theorem 5.4: Assume τ is odd and $\sum_{i=1}^{8} y_i^* = \tau + 1$. Then we have either $\boldsymbol{\alpha}^* \in S_4$ or $0 \le x^* \le \lfloor\lfloor\tau/2\rfloor/2\rfloor$ and

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha}_0 \triangleq (0, 0, \lceil\tau/2\rceil - x^*, x^*, \lceil\tau/2\rceil - x^*, x^*, 0, 0), \tag{5.35}$$

or $0 < x^* \le \lfloor\lfloor\tau/2\rfloor/2\rfloor$ and

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha}_0' \triangleq (0, \lceil\tau/2\rceil - x^*, 0, x^*, \lceil\tau/2\rceil - x^*, 0, x^*, 0). \tag{5.36}$$

Proof: Let $d' \triangleq \lceil\tau/2\rceil - y_1^* - y_2^*$. By (5.29) and $\sum_{i=1}^{8} y_i^* = \tau + 1 = 2\lceil\tau/2\rceil$, we see

$$\begin{cases} y_3^* + y_4^* \ge d', \ y_5^* + y_6^* \ge d', \\ y_1^* + y_3^* + y_5^* + y_7^* \ge \lceil\tau/2\rceil. \end{cases}$$

Clearly, we have $d' \ge \lceil\tau/2\rceil - n^*$, $y_8^* = 0$ and $y_1^* \le \lfloor\tau/2\rfloor$. Furthermore, we conclude $d' \ge 1$. Assume in contrast $d' \le 0$, and thus $y_2^* > 0$. If $y_3^* + y_4^* > 0$, then by $y_5^* + y_6^* \le 2\lceil\tau/2\rceil - (y_1^* + y_2^* + y_3^* + y_4^*) < \lceil\tau/2\rceil$ we see $y_5^* < \lceil\tau/2\rceil - x^*$ or $y_6^* < x^*$ and thus $(y_1^*, y_2^* - 1, y_3^*, y_4^*, y_5^* + 1, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$ or $(y_1^*, y_2^* - 1, y_3^*, y_4^*, y_5^*, y_6^* + 1, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Hence, $y_3^* = y_4^* = 0$. Similarly, we can get $y_5^* = y_6^* = 0$. Thus, from $y_8^* = 0$ and $y_7^* \le x^* \le \lfloor\tau/2\rfloor$ we see $d' < 0$ and then $(y_1^*, y_2^* - 1, y_3^*, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory.

Assume $1 \le d' < \lceil \tau/2 \rceil - x^*$. From $y_8^* = 0$ we see easily

$$y_1^* + y_2^* = \lceil \tau/2 \rceil - d',\ y_3^* + y_4^* = d',\ y_5^* + y_6^* = \lceil \tau/2 \rceil - x^*,\ y_7^* = x^*.$$

If $y_6^* > 0$, then $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^* - 1, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Hence $y_6^* = 0$ and $y_5^* = \lceil \tau/2 \rceil - x^* > 0$. If $y_1^* + y_3^* > 0$, then $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^* - 1, y_6^*, y_7^*, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Hence $y_1^* = y_3^* = 0$ and $y_2^* = \lceil \tau/2 \rceil - d' \le \lceil \tau/2 \rceil - x^*$ and $y_4^* = d' \le x^*$ and thus $1 \le d' = x^* < \lceil \tau/2 \rceil - x^*$, i.e. $0 < x^* \le \lfloor (\lceil \tau/2 \rceil - 1)/2 \rfloor = \lfloor \lfloor \tau/2 \rfloor /2 \rfloor$ and (5.36) holds.

Now we assume $\lceil \tau/2 \rceil - x^* \le d' \le \lceil \tau/2 \rceil$. Clearly, we have

$$y_1^* + y_2^* = y_7^* = \lceil \tau/2 \rceil - d',\ y_3^* + y_4^* = y_5^* + y_6^* = d'.$$

Consider the case of $y_1^* + y_3^* + y_5^* + y_7^* > \lceil \tau/2 \rceil$. If $d' < \lceil \tau/2 \rceil$, then $(y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*, y_7^* - 1, y_8^*) \prec \boldsymbol{\alpha}^*$, contradictory. Hence $d' = \lceil \tau/2 \rceil$ and (5.35) holds and $y_1^* + y_3^* + y_5^* + y_7^* = 2(\lceil \tau/2 \rceil - x^*) > \lceil \tau/2 \rceil$, i.e. $0 \le x^* \le \lfloor \lfloor \tau/2 \rfloor /2 \rfloor$. Now we consider the case of $y_1^* + y_3^* + y_5^* + y_7^* = \lceil \tau/2 \rceil$. For any k with $\lceil \tau/2 \rceil - x^* \le k \le \lceil \tau/2 \rceil$, write

$$\begin{aligned} M(k) &\triangleq \min\{\lceil \tau/2 \rceil - k, n^* + x^* - \lceil \tau/2 \rceil\} + 2(\lceil \tau/2 \rceil - x^*) + (\lceil \tau/2 \rceil - k), \\ m(k) &\triangleq \max\{0, x^* - k\} + 2\max\{0, k - x^*\} + (\lceil \tau/2 \rceil - k). \end{aligned}$$

We can see easily that $M(k) \ge \lceil \tau/2 \rceil$ if and only if

$$k \le \min\{n^* + \lceil \tau/2 \rceil - x^*, \lceil \tau/2 \rceil + \lfloor \lceil \tau/2 \rceil /2 \rfloor - x^*\},$$

and $m(k) \le \lceil \tau/2 \rceil$ if and only if

$$\lceil x^*/2 \rceil \le k \le 2x^*,$$

and thus we see $d_1' \le d' \le d_2'$, i.e. $\boldsymbol{\alpha}^* \in S_4$. □

For any non-empty subset S of Ω_5, we say a pair $\boldsymbol{\alpha} \in S$ is an S-pair if $L^*(\boldsymbol{u}(\boldsymbol{\alpha})) = \min_{\boldsymbol{\alpha}' \in S} L^*(\boldsymbol{u}(\boldsymbol{\alpha}'))$. For convenience, we say φ is a ϕ-pair and define $L^*(\boldsymbol{u}(\varphi)) \triangleq +\infty$. Clearly, similar to the determination of x^*, the number of operations of additions and comparisons of real numbers for determining the S_1-pair and S_2-pair is of order τ. In the next subsection we will give a sub-algorithm_S_3 to determine the S_3-pair and a sub-algorithm_S_4 to determine the

S_4-pair, the numbers of operations of additions and comparisons of real numbers for determining the S_3-pair and S_4-pair are of order τ. Clearly, α_0' also belongs to Ω_5 if $x^* = 0$. Thus by Lemma 5.2 and Theorems 5.3 and 5.4, we have the following algorithm.

Algorithm A

Remark: If the tuple v_4 is given by (5.25), this algorithm gives the tuple v_5.

Input: The integers τ, n, x^*, and the reliability tuple r'.

Output: The tuple v_5.

Step 1 Determine the S_i-pairs α_i for $i = 1, 2, 3$. Let $S_0 \triangleq \{\alpha_1, \alpha_2, \alpha_3\}$.

Step 2 If τ is even, goto Step 4.
Else, goto Step 3.

Step 3 Determine the S_4-pairs α_4, and then put α_4 into S_0.
If $x^* > \lfloor \lfloor \tau/2 \rfloor /2 \rfloor$, goto Step 4.
Else, put α_0 and α_0' into S_0 and goto Step 4.

Step 4 Determine the S_0-pair α^* and output $u(\alpha^*)$ and END. □

Clearly, the number of operations of additions and comparisons of real numbers for determing the n-tuple v_5 is of order τ. (More exactly, $\leq 56\tau$)

5.2.2 Sub-algorithms

Consider to determine the S_3-pair. For any integer k with $d_1^* \leq k \leq d_2^*$, let $S(k)$ denote the set of pairs $\alpha = (y_1, y_2, \ldots, y_8)$ which satisfy (5.27) and (5.32). Write $\kappa_1 \triangleq \tau + 2\lceil \tau/2 \rceil - x^*$, $\kappa_2 \triangleq \tau + \lceil \tau/2 \rceil$, $\kappa_3 \triangleq \tau + x^*$, $\kappa_4 \triangleq \tau$, $\kappa_5 \triangleq \lfloor \tau/2 \rfloor + x^*$, $\kappa_6 \triangleq \lfloor \tau/2 \rfloor$, $\kappa_7 \triangleq \lfloor \tau/2 \rfloor - x^*$, $\kappa_8 \triangleq 0$. For any k with $d_1^* \leq k \leq d_2^*$, let

$$l_1^k \triangleq l_4^k \triangleq \max\{0, x^* - k\},\ l_2^k \triangleq l_3^k \triangleq \max\{0, k - x^*\}, \tag{5.37}$$

$$r_1^k \triangleq \min\{\lceil \tau/2 \rceil - k, n^* + x^* - \lceil \tau/2 \rceil\}, \tag{5.38}$$

$$r_2^k \triangleq r_3^k \triangleq \min\{k, \lceil \tau/2 \rceil - x^*\},\ r_4^k \triangleq \min\{\lfloor \tau/2 \rfloor - k, x^*\}, \tag{5.39}$$

$$\zeta_1^k \triangleq \lceil \tau/2 \rceil - k,\ \zeta_2^k \triangleq \zeta_3^k \triangleq k,\ \zeta_4^k \triangleq \lfloor \tau/2 \rfloor - k. \tag{5.40}$$

By the definitions of d_1^* and d_2^* we see easily

$$\sum_{i=1}^{4} l_i^k \leq \lceil \tau/2 \rceil \leq \sum_{i=1}^{4} r_i^k,\ l_i^k \leq r_i^k,\ i = 1, 2, 3, 4. \tag{5.41}$$

For integers k, i and y with $d_1^* \le k \le d_2^*$ and $1 \le i \le 4$, let

$$s_i^k(y) \triangleq \begin{cases} -\infty, & \text{if } y < l_i^k, \\ 0, & \text{if } y = l_i^k, \\ |r'_{\kappa_{2i-1}+y}| - |r'_{\kappa_{2i}+\zeta_i^k-y+1}|, & \text{if } l_i^k < y \le r_i^k, \\ +\infty, & \text{if } y > r_i^k. \end{cases} \tag{5.42}$$

Clearly, for any integers k and i, $s_i^k(y)$ is monotonously increasing and $s_i^k(y) > 0$ for any y with $y > l_i^k$. Write $\boldsymbol{\alpha}^l \triangleq (l_1^k, \zeta_1^k - l_1^k, l_2^k, \zeta_2^k - l_2^k, l_3^k, \zeta_3^k - l_3^k, l_4^k, \zeta_4^k - l_4^k)$. Then for any pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8) \in S(k)$ we have

$$L^*(\boldsymbol{u}(\boldsymbol{\alpha})) - L^*(\boldsymbol{u}(\boldsymbol{\alpha}^l)) = \sum_{i=1}^{4} \sum_{y=l_i^k}^{y_{2i-1}} s_i^k(y). \tag{5.43}$$

If $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ is an $S(k)$-pair, we see easily that

$$\max_{1 \le i \le 4} \{s_i^k(y_{2i-1})\} \le \min_{1 \le i \le 4} \{s_i^k(y_{2i-1} + 1)\}. \tag{5.44}$$

Furthermore, the following lemma says that a pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ with (5.32) and (5.44) must be an $S(k)$-pair.

Lemma 5.3: Assume integer k satisfies $d_1^* \le k \le d_2^*$. If pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ satisfies (5.32) and (5.44), then $\boldsymbol{\alpha}$ is an $S(k)$-pair.

Proof: At first, we show that $\boldsymbol{\alpha}$ belongs to $S(k)$, i.e. $l_i^k \le y_{2i-1} \le r_i^k$, $i = 1, 2, 3, 4$. Indeed, if $y_{2j-1} < l_j^k$ for some j with $1 \le j \le 4$, by (5.41) we see that there must be a j' with $1 \le j' \le 4$ such that $y_{2j'-1} > l_{j'}^k$ and thus we see $\min_{1 \le i \le 4}\{s_i^k(y_{2i-1} + 1)\} \le 0 < \max_{1 \le i \le 4}\{s_i^k(y_{2i-1})\}$, contradictory. Similarly, if $y_{2j-1} > r_j^k$ for some j with $1 \le j \le 4$, we can get $\max_{1 \le i \le 4}\{s_i^k(y_{2i-1})\} = +\infty > \min_{1 \le i \le 4}\{s_i^k(y_{2i-1}+1)\}$, contradictory.

Since $s_i^k(y)$ is monotonously increasing for any integers k and i, by (5.44) and (5.43) we see easily $L^*(\boldsymbol{u}(\boldsymbol{\alpha})) = \min_{\boldsymbol{\alpha}' \in S(k)} L^*(\boldsymbol{u}(\boldsymbol{\alpha}'))$, i.e. $\boldsymbol{\alpha}$ is a $S(k)$-pair. □

If pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ with (5.32) is not an $S(k)$-pair, then there are two distinct integers $j, j' \in \{1, 2, 3, 4\}$ such that

$$s_j^k(y_{2j-1}) = \max_{1 \le i \le 4}\{s_i^k(y_{2i-1})\} > s_{j'}^k(y_{2j'-1} + 1) = \min_{1 \le i \le 4}\{s_i^k(y_{2i-1} + 1)\}, \tag{5.45}$$

we write

$$\mu(\alpha) \triangleq (y_1 + l_1, y_2 - l_1, y_3 + l_2, y_4 - l_2, y_5 + l_3, y_6 - l_3, y_7 + l_4, y_8 - l_4), \quad (5.46)$$

where $l_j = -1$, $l_{j'} = 1$ and $l_i = 0$ for $i \in \{1,2,3,4\}\backslash\{j, j'\}$. For convenience, we write $\mu^{(0)}(\alpha) \triangleq \alpha$ and $\mu^{(l)}(\alpha) \triangleq \mu(\mu^{(l-1)}(\alpha))$ for any positive integer l. For any i with $1 \leq i \leq 8$, we see easily that the i-th component of $\mu^{(l)}(\alpha)$ is monotonously increasing or monotonously decreasing. Hence, there must be an integer l_0 with $0 \leq l_0 \leq \lceil \tau/2 \rceil$ such that $\mu^{(l_0)}(\alpha)$ is an $S(k)$-pair, or in other words, we have the following lemma.

Lemma 5.4: For any pair $\alpha = (y_1, y_2, \ldots, y_8)$ with (5.32), there must be an integer l_0 with $0 \leq l_0 \leq \lceil \tau/2 \rceil$ such that $\mu^{(l_0)}(\alpha)$ is an $S(k)$-pair.

□

The following lemma shows that we can determine an $S(k-1)$-pair easily from an $S(k)$-pair.

Lemma 5.5: Assume $d_1^* < k \leq d_2^*$. If $\alpha = (y_1, y_2, \cdots, y_8) \in S(k)$ is an $S(k)$-pair, then either $\theta(\alpha) \triangleq (y_1, y_2 + 1, y_3, y_4 - 1, y_5, y_6 - 1, y_7, y_8 + 1)$ or $\mu(\theta(\alpha))$ or $\mu^{(2)}(\theta(\alpha))$ is an $S(k-1)$-pair.

Proof: Clearly, for any integer y we have

$$s_i^{k-1}(y) \leq s_i^k(y) \leq s_i^{k-1}(y+1), \ i = 1, 4, \quad (5.47)$$

$$s_j^{k-1}(y-1) \leq s_j^k(y) \leq s_j^{k-1}(y), \ j = 2, 3. \quad (5.48)$$

Without loss of generality, we assume α^* is not an $S(k-1)$-pair, i.e.

$$X_1 \triangleq \max_{1 \leq i \leq 4}\{s_i^{k-1}(y_{2i-1})\} > X_2 \triangleq \min_{1 \leq i \leq 4}\{s_i^{k-1}(y_{2i-1} + 1)\}. \quad (5.49)$$

1. If $X_1 = s_1^{k-1}(y_1)$, then by (5.49), $s_1^{k-1}(y_1) \leq s_1^{k-1}(y_1 + 1)$ and $s_1^{k-1}(y_1) \leq s_1^k(y_1) \leq \min\{s_2^k(y_3 + 1), s_3^k(y_5 + 1), s_4^k(y_7 + 1)\} \leq \min\{s_2^{k-1}(y_3 + 1), s_3^{k-1}(y_5+1), s_4^{k-1}(y_7+2)\}$ we see $X_2 = s_4^{k-1}(y_7+1)$ and $\min\{s_1^{k-1}(y_1), s_2^{k-1}(y_3+1), s_3^{k-1}(y_5 + 1), s_4^{k-1}(y_7 + 2)\} = s_1^{k-1}(y_1)$. Thus by $s_1^{k-1}(y_1) \geq \max\{s_1^{k-1}(y_1 - 1), s_2^{k-1}(y_3), s_3^{k-1}(y_5), s_4^{k-1}(y_7 + 1)\}$ and Lemma 5.3 we see $\mu(\alpha) = (y_1 - 1, y_2 + 2, y_3, y_4 - 1, y_5, y_6 - 1, y_7 + 1, y_8)$ is a $S(k-1)$-pair.

2. If $X_1 = s_4^{k-1}(y_7)$, similar to 1, we can show that $\mu(\alpha) = (y_1+1, y_2, y_3, y_4 - 1, y_5, y_6 - 1, y_7 - 1, y_8 + 2)$ is a $S(k-1)$-pair.

3. If $X_2 = s_2^{k-1}(y_3+1)$, then by (5.49), $s_2^{k-1}(y_3+1) \geq s_2^{k-1}(y_3)$ and $s_2^{k-1}(y_3 + 1) \geq s_2^k(y_3+1) \geq \max\{s_1^k(y_1), s_4^k(y_7), s_3^k(y_5)\} \geq \max\{s_1^{k-1}(y_1), s_4^{k-1}(y_7), s_3^{k-1}(y_5 - 1)\}$ we see $X_1 = s_3^{k-1}(y_5)$ and $\max\{s_1^{k-1}(y_1), s_2^{k-1}(y_3+1), s_3^{k-1}(y_5-1), s_4^{k-1}(y_7)\} = s_2^{k-1}(y_3+1)$. Thus by $s_2^{k-1}(y_3+1) \leq \min\{s_1^{k-1}(y_1+1), s_2^k(y_3+2), s_3^{k-1}(y_5), s_4^{k-1}(y_7+1)\}$ and Lemma 5.3 we see $\mu(\alpha) = (y_1, y_2+1, y_3+1, y_4-2, y_5, y_6-1, y_7, y_8+1)$ is a $S(k-1)$-pair.

4. If $X_2 = s_3^{k-1}(y_5+1)$, similar to the result 3, we can show that $\mu(\alpha) = (y_1, y_2+1, y_3-1, y_4, y_5+1, y_6-2, y_7, y_8+1)$ is a $S(k-1)$-pair.

5. Now we consider the case of

$$X_1 = \max\{s_2^{k-1}(y_3), s_3^{k-1}(y_5)\},\ X_2 = \min\{s_1^{k-1}(y_1+1), s_4^{k-1}(y_7+1)\}. \quad (5.50)$$

Without loss of generality, assume $X_1 = s_2^{k-1}(y_3)$ and $X_2 = s_1^{k-1}(y_1+1)$, i.e. $\mu(\alpha) = (y_1+1, y_2, y_3-1, y_4, y_5, y_6-1, y_7, y_8+1)$. Let

$$X_1^* \triangleq \max\{s_1^{k-1}(y_1+1), s_2^{k-1}(y_3-1), s_3^{k-1}(y_5), s_4^{k-1}(y_7)\}, \quad (5.51)$$
$$X_2^* \triangleq \min\{s_1^{k-1}(y_1+2), s_2^{k-1}(y_3), s_3^{k-1}(y_5+1), s_4^{k-1}(y_7+1)\}. \quad (5.52)$$

If $X_1^* \leq X_2^*$, then by Lemma 5.3 we see $\mu(\alpha)$ is a $S(k-1)$-pair. Now we assume $X_1^* > X_2^*$. Clearly, we have $s_2^{k-1}(y_3) \geq X_1^*$ and $s_1^{k-1}(y_1+1) \leq X_2^*$. By

$$\begin{aligned} s_3^{k-1}(y_5+1) \geq s_3^k(y_5+1) \geq s_2^k(y_3) \geq s_2^{k-1}(y_3-1), \\ s_3^{k-1}(y_5+1) \geq s_3^k(y_5+1) \geq s_4^k(y_7) \geq s_4^{k-1}(y_7), \\ s_3^{k-1}(y_5+1) \geq s_1^{k-1}(y_1+1),\ s_2^{k-1}(y_3) \geq s_4^{k-1}(y_7), \\ s_1^{k-1}(y_1+2) \geq s_1^k(y_1+1) \geq s_4^k(y_7) \geq s_4^{k-1}(y_7), \end{aligned}$$

we see $s_3^{k-1}(y_5+1) \geq X_1^*$ and $s_4^{k-1}(y_7) \leq X_2^*$. Hence, by $X_1^* > X_2^*$ we have

$$X_1^* = \max\{s_2^{k-1}(y_3-1), s_3^{k-1}(y_5)\},\ X_2^* = \min\{s_1^{k-1}(y_1+2), s_4^{k-1}(y_7+1)\} (5.53)$$

If $X_1^* = s_2^{k-1}(y_3-1)$, from $\min\{s_1^{k-1}(y_1+2), s_4^{k-1}(y_7+2), s_3^{k-1}(y_5+1)\} \geq \min\{s_1^k(y_1+1), s_4^k(y_7+1), s_3^k(y_5+1)\} \geq s_2^k(y_3) \geq s_2^{k-1}(y_3-1)$, we see $X_2^* = s_4^k(y_7 + 1)$ and $\min\{s_1^{k-1}(y_1+2), s_2^{k-1}(y_3-1), s_3^{k-1}(y_5+1), s_4^{k-1}(y_7+2)\} = s_2^{k-1}(y_3-1)$. Thus by $s_2^{k-1}(y_3-1) \geq \max\{s_1^{k-1}(y_1+1), s_2^{k-1}(y_3-2), s_3^{k-1}(y_5), s_4^{k-1}(y_7+1)\}$ and

Lemma 5.3 we see $\mu^{(2)}(\boldsymbol{\alpha}) = (y_1+1, y_2, y_3-2, y_4+1, y_5, y_6-1, y_7+1, y_8)$ is a $S(k-1)$-pair.

If $X_2^* = s_1^{k-1}(y_1+2)$, from $s_1^{k-1}(y_1+2) \geq s_2^{k-1}(y_3-1)$ we see $X_1^* = s_3^{k-1}(y_5)$. Furthermore, from $\max\{s_4^{k-1}(y_7), s_2^{k-1}(y_3-1)\} \leq \max\{s_4^k(y_7), s_2^k(y_3)\} \leq s_1^k(y_1+1) \leq s_1^{k-1}(y_1+2)$, we see $\max\{s_1^{k-1}(y_1+2), s_2^{k-1}(y_3-1), s_3^{k-1}(y_5-1), s_4^{k-1}(y_7)\} = s_1^{k-1}(y_1+2)$. Thus by $s_1^{k-1}(y_1+2) \leq \min\{s_1^{k-1}(y_1+3), s_2^{k-1}(y_3), s_3^{k-1}(y_5), s_4^{k-1}(y_7+1)\}$ and Lemma 5.3 we see $\mu^{(2)}(\boldsymbol{\alpha}) = (y_1+2, y_2-1, y_3-1, y_4, y_5-1, y_6, y_7, y_8+1)$ is a $S(k-1)$-pair.

If $X_1^* = s_3^{k-1}(y_5)$ and $X_2^* = s_4^{k-1}(y_7+1)$, from $X_1 > X_2$ and $X_1^* > X_2^*$ we can also easily prove that

$$\begin{aligned}&\max\{s_1^{k-1}(y_1+1), s_2^{k-1}(y_3-1), s_3^{k-1}(y_5-1), s_4^{k-1}(y_7+1)\}\\ &\leq \min\{s_1^{k-1}(y_1+2), s_2^{k-1}(y_3), s_3^{k-1}(y_5), s_4^{k-1}(y_7+2)\}.\end{aligned}$$

Thus by Lemma 5.3 we see $\mu^{(2)}(\boldsymbol{\alpha}) = (y_1+1, y_2, y_3-1, y_4, y_5-1, y_6, y_7+1, y_8)$ is a $S(k-1)$-pair. □

By Lemmas 5.3, 5.4 and 5.5, we can determine the S_3-pair $\boldsymbol{\alpha}_3$ by using of the following sub-algorithm.

Sub-algorithm_S_3

Input: The integers τ, n, x^*, and the reliability tuple $\boldsymbol{r}'$.

Output: The S_3-pair $\boldsymbol{\alpha}_3$.

Step 1 Compute d_1^* and d_2^*.
  If $d_1^* > d_2^*$, output φ and END.
  Else, goto Step 2.

Step 2 Set $k = d_2^*$ and select an arbitrary pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ such that (5.32) holds and goto Step 3.

Step 3 If (5.44) holds, goto Step 4.
  Else, replace the pair $\boldsymbol{\alpha}$ by $\mu(\boldsymbol{\alpha})$, and goto Step 3.

Step 4 Let $\boldsymbol{\alpha}'' = \boldsymbol{\alpha}$, $\ell' = \ell'' = L^*(\boldsymbol{u}(\boldsymbol{\alpha}))$ and goto Step 5.

Step 5 If $k = d_1^*$, output $\boldsymbol{\alpha}''$ and END.
  Else, replace k and $\boldsymbol{\alpha}$ by $k-1$ and $\theta(\boldsymbol{\alpha})$ respectively.
    Replace ℓ' by $L^*(\boldsymbol{u}(\boldsymbol{\alpha}))$ and goto Step 6.

Step 6 If (5.44) holds, goto Step 7.
  Else, replace $\boldsymbol{\alpha}$ by $\mu(\boldsymbol{\alpha})$, and then replace ℓ' by $L^*(\boldsymbol{u}(\boldsymbol{\alpha}))$.

If (5.44) holds, goto Step 7.

Else, replace $\boldsymbol{\alpha}$ by $\mu(\boldsymbol{\alpha})$, replace ℓ' by $L^*(\boldsymbol{u}(\boldsymbol{\alpha}))$ and goto Step 7.

Step 7 If $\ell'' \leq \ell'$, goto Step 5.

Else, replace $\boldsymbol{\alpha}''$ and ℓ'' by $\boldsymbol{\alpha}$ and ℓ' respectively and goto Step 5. □

Remark: In the procedure of getting the $S(k-1)$-pair from the $S(k)$-pair, $\boldsymbol{\alpha}''$ expresses the best pair in $\cup_{k'=k}^{d_2^*} S(k')$, and $\ell'' = L^*(\boldsymbol{u}(\boldsymbol{\alpha}''))$.

Consider the computational complexity of this sub-algorithm. From a pair $\boldsymbol{\alpha}$ to determine the pair $\mu(\boldsymbol{\alpha})$, it needs 15 operations of additions and comparisons of real numbers. Hence, to get the $S(d_2^*)$-pair, it needs at most $15\lceil\tau/2\rceil$ operations of additions and comparisons of real numbers. From the $S(k)$-pair to get the $S(k-1)$-pair and the best pair in $\cup_{k'=k-1}^{d_2^*} S(k')$, it needs at most $2\times15+(12+1) = 43$ operations of additions and comparisons of real numbers. Hence, the total number of operations of additions and comparisons of real numbers for this sub-algorithm is not more than $15\lceil\tau/2\rceil + \tau + 43(d_2^* - d_1^*) \leq 30\tau$.

The above method can also be used to determine the S_4-pair. For any integer d' with $d_1' \leq k \leq d_2'$, let $S'(k)$ denote the set of pairs $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ which satisfy (5.27) and (5.34). The same method in the proof of Lemma 5.3 can deduce the following analogue for $S'(k)$.

Lemma 5.6: Assume integer k satisfies $d_1' \leq k \leq d_2'$. A pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ satisfying (5.34) is an $S'(k)$-pair if and only if

$$\max_{1\leq i\leq 3}\{s_i^k(y_{2i-1})\} \leq \min_{1\leq i\leq 3}\{s_i^k(y_{2i-1}+1)\}. \tag{5.54}$$

□

If pair $\boldsymbol{\alpha} = (y_1, y_2, \ldots, y_8)$ with (5.34) is not $S'(k)$-pair, then there are two distinct integers $j, j' \in \{1,2,3\}$ such that

$$s_j^k(y_{2j-1}) = \max_{1\leq i\leq 3}\{s_i^k(y_{2i-1})\} > s_{j'}^k(y_{2j'-1}+1) = \min_{1\leq i\leq 3}\{s_i^k(y_{2i-1}+1)\}, \tag{5.55}$$

we write

$$\hat{\mu}(\boldsymbol{\alpha}) \triangleq (y_1+l_1, y_2-l_1, y_3+l_2, y_4-l_2, y_5+l_3, y_6-l_3, y_7, y_8), \tag{5.56}$$

where $l_j = -1$, $l_{j'} = 1$ and $l_i = 0$ for $i \in \{1,2,3\}\backslash\{j,j'\}$. Write $\hat{\mu}^{(0)}(\boldsymbol{\alpha}) \triangleq \boldsymbol{\alpha}$ and $\hat{\mu}^{(l)}(\boldsymbol{\alpha}) \triangleq \hat{\mu}(\hat{\mu}^{(l-1)}(\boldsymbol{\alpha}))$ for any positive integer l. We see easily that there is an integer l_0' with $0 \leq l_0' \leq k$ such that $\hat{\mu}^{(l_0')}(\boldsymbol{\alpha})$ is an $S'(k)$-pair, or in other words, we have the following lemma.

Lemma 5.7: For any pair $\alpha = (y_1, y_2, \ldots, y_8)$ with (5.34), there is an integer l'_0 with $0 \le l'_0 \le k$ such that $\hat{\mu}^{(l'_0)}(\alpha)$ is an $S'(k)$-pair.

Similar to Lemma 5.5, we have the following lemma.

Lemma 5.8: Assume $d'_1 < k \le d'_2$. If $\alpha = (y_1, y_2, \ldots, y_8) \in S'(k)$ is an $S'(k)$-pair and $s^{k-1}_{i_0}(y_{2i_0-1}) = \max_{1 \le i \le 3} s^{k-1}_i(y_{2i-1})$, then either $\theta'(\alpha) \triangleq (y_1 - l'_1, y_2 + 1 + l'_1, y_3 - l'_2, y_4 - 1 + l'_2, y_5 - l'_3, y_6 - 1 + l'_3, y_7 + 1, y_8)$ or $\hat{\mu}(\theta'(\alpha))$ is an $S'(k-1)$-pair, where $l_{i_0} = 1$ and $l_i = 0$ for $i \in \{1, 2, 3\} \backslash \{i_0\}$.

□

By Lemmas 5.6, 5.7 and 5.8, we can also determine the S_4-pair by the following analogue of Sub-algorithm_S_3.

Sub-algorithm_S_4
Input: The integers τ, n, x^*, and the reliability tuple $\boldsymbol{r}'$.
Output: The S_4-pair $\boldsymbol{\alpha}_4$.
Step 1 Compute d'_1 and d'_2.
 If $d'_1 > d'_2$, output φ and END.
 Else, goto Step 2.
Step 2 Set $k = d'_2$ and select an arbitrary pair $\alpha = (y_1, y_2, \ldots, y_8)$ such that (5.34) holds and goto Step 3.
Step 3 If (5.54) holds, goto Step 4.
 Else, replace the pair α by $\hat{\mu}(\alpha)$ and goto Step 3.
Step 4 Let $\alpha'' = \alpha$, $\ell' = \ell'' = L^*(\boldsymbol{u}(\alpha))$ and goto Step 5.
Step 5 If $k = d'_1$, output α'' and END.
 Else, replace k and α by $k-1$ and $\theta'(\alpha)$ respectively.
  Replace ℓ' by $L^*(\boldsymbol{u}(\alpha))$ and goto Step 6.
Step 6 If (5.54) holds, goto Step 7.
 Else, replace α by $\hat{\mu}(\alpha)$.
  Replace ℓ' by $L^*(\boldsymbol{u}(\alpha))$ and goto Step 7.
Step 7 If $\ell'' \le \ell'$, goto Step 5.
 Else, replace α'' and ℓ'' by α and ℓ' respectively and goto Step 5. □

We see easily that the number of operations of additions and comparisons of real numbers for this sub-algorithm is of order τ too. Indeed, it is not difficult to show that the number of operations of additions and comparisons of real numbers for this sub-algorithm is not more than 18τ.

Chapter 6

Conclusions

In this thesis, optimality conditions and ruling-out conditions for iterative soft-decision decoding algorithms are investigated.

At first, a sufficient condition, denoted $\text{Cond}_{\text{opt},h}$, for the optimality of a decoded codeword based on h reference words are derived for any iterative soft-decision decoding algorithms which are based on the generation of a series of codeword candidates.

Next, for a class of Chase-type decoding algorithms which consist of a series of bounded distance decodings that are implemented around some words obtained by adding a given word to the error patterns whose nonzero components are confined to those bit positions with less reliability, some effective optimality conditions and ruling-out conditions are devised. For some generalizations of Chase-type algorithm and the minimum weight trellis search (MWTS) decoding algorithm and some generalizations of GMD decoding, some optimality conditions are investigated.

Then, these testing conditions are formulated uniformly into the optimal value of a class of nonlinear integer programming problems (IPPs), which can be used in many iterative soft-decision decoding algorithms. The IPPs are solved for some particular cases. Especially, the IPP related to the optimality condition $\text{Cond}_{\text{opt},h}$ with $h \leq 5$ is split into a few simpler sub-IPPs, the number of variables of each sub-IPP is only half of that of the original IPP. Some effective algorithms for solving the IPP related to the optimality condition $\text{Cond}_{\text{opt},h}$ with $h \leq 4$ are given. For the computational complexity of these algorithms, the number of operations of additions and comparisons of real number of the algorithms is shown to be of order N for $h \leq 3$, and of order N^2 for $h = 4$, where N is the length of the code. Thus, the optimality condition $\mathrm{Cond}_{\mathrm{opt},h}$ with $h \leq 4$ can reduce significantly the average decoding complexity and provide fast termination of the decoding iteration without degrading the error performance.

By use of partial information of the distance profile of the code, an improvement of the optimality condition $\mathrm{Cond}_{\mathrm{opt},h}$ is presented further. Some methods for evaluating the improved optimality condition for the simplest cases are shown.

Finally, for a multistage Chase-type decoding algorithm that consists of a series of Chase-type decodings $\mathrm{Chase}(\boldsymbol{v}^{(j)}, t_0, \tau_0)$ for which the next search center is chosen as the best one among the words that have not been visited, two algorithms for determining the fourth and fifth search centers are presented, respectively. The numbers of operations of additions and comparisons of real numbers for these two algorithms are of order $\tau (= t_0 + 1)$.

References

[1] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Trans. Inform. Theory, vol.IT-18, no.1, pp.170-182, Jan. 1972.

[2] G. D. Forney, Jr., "Generalized Minimum Distance Decoding," IEEE Trans. Inform. Theory, vol.IT-12, no.2, pp.125-131, April 1966.

[3] M. P. C. Fossorier and S. Lin "Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics," IEEE Trans. Inform. Theory, vol.41, no.5, pp.1379–1395, Sept. 1995.

[4] M. P. C. Fossorier and S. Lin, "Complementary Reliability-Based Decodings of Binary Linear Block Codes," IEEE Trans. Inform. Theory, vol.43, no.5, pp.1667–1672, Sept. 1997.

[5] M. P. C. Fossorier and S. Lin "A Unified Method for Evaluating the Error-Correction Radius of Reliability-Based Soft-Decision Algorithms for Linear Block Codes," IEEE Trans. Inform. Theory, vol.44, no.2, pp.691–670, March 1998.

[6] M. P. C. Fossorier and S. Lin "Generalized Chase and GMD Decodings," Proc. Int. Symp. Inform. Theory & Appl., Mexico City, Mexico, pp.415–418, Oct. 1998.

[7] M. P. C. Fossorier, S. Lin and D. Rhee, "Bit-Error Probability for Maximum-likelihood Decoding of Linear Block Codes and Related Soft-Decision Decoding Methods," IEEE Trans. Inform. Theory, vol.44, no.6, pp.3083–3090, Nov. 1998.

[8] J. Hagenauer, E. Offer and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," IEEE Trans. Inform. Theory, vol.42, no.2, pp.429–445, March 1996.

[9] Y. S. Han, C. R. P. Hartmann and C. C. Chen, "Efficient Priority-First Search Maximum-Likelihood Soft-Decision Decoding of Linear Block Codes," IEEE Trans. Inform. Theory, vol.39, no.5, pp.1514–1523, Sept. 1993.

[10] N. Kamiya, "On Acceptance Criterion for Efficient Successive Errors-and-Erasures Decoding of Reed-Solomon and BCH Codes," IEEE Trans. Inform. Theory, vol.43, no.5, pp.1477–1488, Sept. 1997.

[11] T. Kaneko, T. Nishijima and S. Hirasawa, "An Improvement of Soft-Decision Maximum-Likelihood Decoding Algorithm Using Hard-Decision Bounded-Distance Decoding," IEEE Trans. Inform. Theory, vol.43, no.4, pp.1314–1319, July 1997.

[12] T. Kaneko, T. Nishijima, H. Inazumi and S. Hirasawa, "An Efficient Maximum Likelihood Decoding Algorithm for Linear Block Codes with Algebraic Decoder," IEEE Trans. Inform. Theory, vol.40, no.2, pp.320-327, March 1994.

[13] T. Kasami, "On Integer Programming Problems Related to Soft-Decision Iterative Decoding Algorithms," Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: Proceedings of the 13th International Symposium, AAECC-13, Honolulu, Hawaii, USA, Lecture Notes in Computer Science, vol.1719, pp.43–54, Spring Verlag, November 1999.

[14] T. Kasami and T. Koumoto, "Computational Complexity for Computing Sufficient Conditions on the Optimality of a Decoded Codeword," Technical Report of NAIST, NAIST-IS-TR98008, July, 1998.

[15] T. Kasami, T. Koumoto, T. Takata and S. Lin, "The Effectiveness of the Least Stringent Sufficient Condition on Optimality of Decoded Codewords," Proc. 3rd Int. Symp. Commun. Theory & Appl., Ambleside, UK, pp.324-333, July 1995.

[16] T. Kasami, T. Takata, T. Koumoto, T. Fujiwara, H. Yamamoto and S. Lin, "The Least Stringent Sufficient Condition on Optimality of Suboptimal Decoded Codewords," Technical Report of IEICE, Japan, IT94-82, Jan. 1995.

[17] T. Koumoto and T. Kasami, "An Iterative Decoding Algorithm Based on Information of Decoding Failure," Proc. 20th Symp. Inform. Theory & Appl., Matsuyama, Japan, pp.325-328, Dec. 1997.

[18] T. Koumoto and T. Kasami, "Analysis and Improvement on GMD Like Decoding Algorithms," Proc. Int. Symp. Inform. Theory & Appl., Mexico City, Mexico, pp.419–422, Oct. 1998.

[19] T. Koumoto and T. Kasami, "Analysis and Improvement on Chase Type 2 Like Decoding Algorithms," Proc. 21st Symp. Inform. Theory & Appl., pp.531–534, Gifu, Japan, Dec. 1998.

[20] T. Koumoto, T. Kasami and S. Lin, "A sufficient Condition for Ruling Out Some Useless Test Error Patterns in Iterative Decoding Algorithms," IEICE Trans. Fundamentals, vol.E81-A, no.2, pp.321–326, Feb. 1998.

[21] T. Koumoto, T. Takata, T. Kasami and S. Lin, "A Low-Weight Trellis Based Iterative Soft-Decision Decoding Algorithm for Binary Linear Block Codes," IEEE Trans. Inform. Theory, vol.45, no.2, pp.731–741, March 1999.

[22] S. Lin, H. T. Moorthy and T. Kasami, "An Efficient Soft-Decision Decoding Scheme for Binary Linear Block Codes," Proc. 3rd Int. Symp. Commun. Theory & Appl., Ambleside, UK, pp.4-11, July 1995.

[23] H. T. Moorthy, S. Lin and T. Kasami, "Soft-Decision Decoding of Binary Linear Block Codes Based on an Iterative Search Algorithm," IEEE Trans. Inform. Theory, vol.43, no.2, pp.1030–1040, May 1997.

[24] B. Z. Shen, K. K. Tzeng and C. Wang, "A Bounded-Distance Decoding Algorithm for Binary Linear Block Codes Achieving the Minimum Effective Error Coefficient," IEEE Trans. Inform. Theory, vol.42, no.6, pp.1987–1991, Nov. 1996.

[25] D. J. Taipale and M. B. Pursley, "An Improvement to Generalized Minimum-Distance Decoding," IEEE Trans. Inform. Theory, vol.37, no.1, pp.167-172, Jan. 1991.

[26] H. Tanaka and Kakigahara, "Simplified Correlation Decoding by Selecting Possible Codewords Using Erasure Information," IEEE Trans. Inform. Theory, vol.IT-29, no.5, pp.743–748, Sept. 1983.

[27] N. Tendolar and C. P. Hartmann, "Generalization of Chase Algorithms for Soft-Decision Decoding of Binary Linear Codes," IEEE Trans. Inform. Theory, vol.IT-30, no.5, pp.714–721, Sept. 1984.

Appendix

A Proof of Lemma 2.5

To prove Lemma 2.5, we first show the following lemma.

Lemma A.1: For any tuple $\boldsymbol{u}' \in \mathcal{T}_j(\boldsymbol{v})$ and $1 \le i < j$,

$$\begin{aligned} &L_{2i-p,2i-p+1}(\boldsymbol{u}') - L_{2i-p,2i-p+1}(\boldsymbol{u}^*) \\ &\ge (d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) - d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v}))|r_{2i-p}|. \end{aligned} \tag{A.1}$$

Proof: If $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) = d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v})$, from the definition of $\boldsymbol{u}^*$ we must have $L_{2i-p,2i-p+1}(\boldsymbol{u}') \ge L_{2i-p,2i-p+1}(\boldsymbol{u}^*)$ and thus (A.1) holds.

If $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) < d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v})$, then from

$$L_{2i-p,2i-p+1}(\boldsymbol{u}^*) \le |r_{2i-p}|$$

we see that

$$\begin{aligned} &L_{2i-p,2i-p+1}(\boldsymbol{u}') - L_{2i-p,2i-p+1}(\boldsymbol{u}^*) \ge -|r_{2i-p}| \\ &\ge (d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) - d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v}))|r_{2i-p}|, \end{aligned}$$

i.e. (A.1) holds.

Now we consider the case of $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) > d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v})$.

If $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v}) = 0$, from the definition of $\boldsymbol{u}^*$ we can show easily that $v_{2i-p} = z_{2i-p}$ and $v_{2i-p+1} = z_{2i-p+1}$, and thus (A.1) holds.

The remaining case is that $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}^*,\boldsymbol{v}) = 1$ and $d_{\mathrm{H},2i-p,2i-p+1}(\boldsymbol{u}',\boldsymbol{v}) = 2$. From the definition of $\boldsymbol{u}^*$, we can show easily that either $v_{2i-p} = z_{2i-p}$ or $v_{2i-p+1} = z_{2i-p+1}$ holds. Furthermore, if $v_{2i-p} = z_{2i-p}$ and $v_{2i-p+1} = z_{2i-p+1}$, then

$$u^*_{2i-p} \ne z_{2i-p},\ u^*_{2i-p+1} = z_{2i-p+1},\ u'_{2i-p} \ne z_{2i-p},\ u'_{2i-p+1} \ne z_{2i-p+1},$$

and thus (A.1) holds. If $v_{2i-p} \ne z_{2i-p}$ and $v_{2i-p+1} = z_{2i-p+1}$, then

$$u^*_{2i-p} = z_{2i-p},\ u^*_{2i-p+1} = z_{2i-p+1},\ u'_{2i-p} = z_{2i-p},\ u'_{2i-p+1} \ne z_{2i-p+1},$$

this also implies (A.1). If $v_{2i-p} = z_{2i-p}$ and $v_{2i-p+1} \ne z_{2i-p+1}$, then

$$u^*_{2i-p} = z_{2i-p},\ u^*_{2i-p+1} = z_{2i-p+1},\ u'_{2i-p} \ne z_{2i-p},\ u'_{2i-p+1} = z_{2i-p+1},$$

and (A.1) holds too. □

Proof of Lemma 2.5: $\boldsymbol{u}^* \in \mathcal{T}_j(\boldsymbol{v})$ follows from (2.83) and (2.86).

Below we will prove by induction that

$$L_{2j'-p,N}(\boldsymbol{u}^*) = \underline{L_{2j'-p,N}}[\mathcal{T}_j(\boldsymbol{v})] \tag{A.2}$$

holds for $j' = j, j-1, \ldots, 1$, and thus from $u_i^* = z_i$ for $1 \le i < 2-p$ we see that $\boldsymbol{u}^*$ satisfies (2.75).

From (2.77) and (2.84), we see that (A.2) holds for $j' = j$.

Now we assume that (A.2) holds for $j' = j^* + 1$ with $1 \le j^* < j$. We want to prove (A.2) holds for $j' = j^*$.

If either $f_{j^*+1} > \rho - j^*$ or $v_{2j^*-p} \ne z_{2j^*-p}$ or $v_{2j^*-p+1} \ne z_{2j^*-p+1}$ holds, then $u^*_{2j^*-p} = z_{2j^*-p}$ and $u^*_{2j^*-p+1} = z_{2j^*-p+1}$, and thus from

$$\begin{aligned}\underline{L_{2j^*-p,N}}[\mathcal{T}_j(\boldsymbol{v})] \le L_{2j^*-p,N}(\boldsymbol{u}^*) &= L_{2(j^*+1)-p,N}(\boldsymbol{u}^*)\\ &= \underline{L_{2(j^*+1)-p,N}}[\mathcal{T}_j(\boldsymbol{v})] \le \underline{L_{2j^*-p,N}}[\mathcal{T}_j(\boldsymbol{v})],\end{aligned}$$

we see that (A.2) holds for $j' = j^*$.

The remaining case is that the following equalities

$$f_{j^*+1} = \rho - j^*, \tag{A.3}$$

$$v_{2j^*-p} = z_{2j^*-p} \text{ and } v_{2j^*-p+1} = z_{2j^*-p+1}, \tag{A.4}$$

hold. For this case, we have

$$u^*_{2j^*-p} \ne z_{2j^*-p} \text{ and } u^*_{2j^*-p+1} = z_{2j^*-p+1}. \tag{A.5}$$

Next we will deduce a contradiction if (A.2) does not hold for $j' = j^*$.

Assume that (A.2) does not hold for $j' = j^*$ and $\boldsymbol{u}'$ is a tuple in $\bigcap_{i=j^*}^{j} T(i, \rho - i + 1, \boldsymbol{v})$ for which

$$L_{2j^*-p,N}(\boldsymbol{u}') = \underline{L_{2j^*-p,N}}[\textstyle\bigcap_{i=j^*}^{j} T(i, \rho - i + 1, \boldsymbol{v})] < L_{2j^*-p,N}(\boldsymbol{u}^*). \tag{A.6}$$

If either $u'_{2j^*-p} \ne z_{2j^*-p}$ or $u'_{2j^*-p+1} \ne z_{2j^*-p+1}$ holds, then from (A.4) and (A.5)

$$\begin{aligned}L_{2j^*-p,N}(\boldsymbol{u}^*) &= |r_{2j^*-p}| + L_{2(j^*+1)-p,N}(\boldsymbol{u}^*)\\ &\le |r_{2j^*-p}| + L_{2(j^*+1)-p,N}(\boldsymbol{u}') \le L_{2j^*-p,N}(\boldsymbol{u}').\end{aligned}$$

contradicts (A.6). Hence

$$u'_{2j^*-p} = z_{2j^*-p} \text{ and } u'_{2j^*-p+1} = z_{2j^*-p+1}, \tag{A.7}$$

and thus from (A.4) and $\boldsymbol{u}' \in T(j^*, \rho - j^* + 1, \boldsymbol{v})$ we get

$$d_{\mathrm{H},2(j^*+1)-p,N}(\boldsymbol{u}', \boldsymbol{v}) = d_{\mathrm{H},2j^*-p,N}(\boldsymbol{u}', \boldsymbol{v}) \geq \rho - j^* + 1. \tag{A.8}$$

If there is an integer i with $j^* + 1 < i \leq j$ such that

$$d_{\mathrm{H},2i-p,N}(\boldsymbol{u}', \boldsymbol{v}) \leq d_{\mathrm{H},2i-p,N}(\boldsymbol{u}^*, \boldsymbol{v}),$$

then the tuple $\boldsymbol{u}''$ with

$$p_{1,2i-p-1}(\boldsymbol{u}'') = p_{1,2i-p-1}(\boldsymbol{u}') \text{ and } p_{2i-p,N}(\boldsymbol{u}'') = p_{2i-p,N}(\boldsymbol{u}^*)$$

is also in $\mathcal{T}_j(\boldsymbol{v})$ and satisfies

$$\begin{aligned} L_{2j^*-p,N}(\boldsymbol{u}'') &= L_{2j^*-p,2i-p-1}(\boldsymbol{u}') + L_{2i-p,N}(\boldsymbol{u}^*) \\ &\leq L_{2j^*-p,2i-p-1}(\boldsymbol{u}') + L_{2i-p,N}(\boldsymbol{u}') = L_{2j^*-p,N}(\boldsymbol{u}'). \end{aligned}$$

Hence, without loss of generality, we may assume that
Case 1:

$$d_{\mathrm{H},2j'-p,N}(\boldsymbol{u}', \boldsymbol{v}) > d_{\mathrm{H},2j'-p,N}(\boldsymbol{u}^*, \boldsymbol{v}) \tag{A.9}$$

holds for $j^* + 1 \leq j' \leq j$, or
Case 2: there is an integer j_0 with $j^* + 1 \leq j_0 < j$ such that (A.9) holds for $j^* + 1 \leq j' \leq j_0$ and

$$u'_i = u^*_i, \text{ for } i \geq 2(j_0 + 1) - p. \tag{A.10}$$

For the Case 1, from (2.80) and (A.9) we have

$$\begin{aligned} L_{2j-p,N}(\boldsymbol{u}') &\geq \underline{L_{2j-p,N}}[T(j, d_{\mathrm{H},2j-p,N}(\boldsymbol{u}', \boldsymbol{v}), \boldsymbol{v})] \\ &\geq \underline{L_{2j-p,N}}[T(j, d_{\mathrm{H},2j-p,N}(\boldsymbol{u}^*, \boldsymbol{v}), \boldsymbol{v})] \\ &\qquad +(d_{\mathrm{H},2j-p,N}(\boldsymbol{u}', \boldsymbol{v}) - d_{\mathrm{H},2j-p,N}(\boldsymbol{u}^*, \boldsymbol{v}))|r_{2j-p}| \\ &= L_{2j-p,N}(\boldsymbol{u}^*) + (d_{\mathrm{H},2j-p,N}(\boldsymbol{u}', \boldsymbol{v}) - d_{\mathrm{H},2j-p,N}(\boldsymbol{u}^*, \boldsymbol{v}))|r_{2j-p}|, \end{aligned} \tag{A.11}$$

and thus from (A.1) and (A.9) we can see easily that

$$L_{2i-p,N}(\boldsymbol{u}') - L_{2i-p,N}(\boldsymbol{u}^*) \geq (d_{\mathrm{H},2i-p,N}(\boldsymbol{u}', \boldsymbol{v}) - d_{\mathrm{H},2i-p,N}(\boldsymbol{u}^*, \boldsymbol{v}))|r_{2i-p}| \quad \text{(A.12)}$$

holds for all i with $j^* + 1 \leq i \leq j$. Hence, by (A.3), (A.4), (A.7), (A.8) and (A.12), we see that

$$\begin{aligned} L_{2j^*-p,N}(\boldsymbol{u}') &= L_{2(j^*+1)-p,N}(\boldsymbol{u}') \\ &\geq L_{2(j^*+1)-p,N}(\boldsymbol{u}^*) + |r_{2j^*+1-p}| \geq L_{2j^*-p,N}(\boldsymbol{u}^*), \end{aligned} \quad \text{(A.13)}$$

contradicts (A.6).

For the Case 2, from (A.1), (A.9) and (A.10), we see also that (A.12) holds for all i with $j^* + 1 \leq i \leq j_0$ and thus (A.13) holds too, it contradicts (A.6). □

B Proof of Lemma 4.4

For $\boldsymbol{\gamma} \in \Gamma_0^h$, let $\zeta(\boldsymbol{\gamma}) \triangleq \{i \in [1, h] : \gamma_i \leq 1\}$. If $p_{\zeta(\boldsymbol{\gamma})}[\Xi]$ is an $m(p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\gamma}))$-set, then Ξ is an $m(\boldsymbol{\gamma})$-set. To prove Lemma 4.4, we give a lemma at first.

Lemma B.1: For $h \leq 5$ and $\boldsymbol{\gamma} \in \Gamma_0^h$, a set $\Xi \subset B^h$ is an $m(\boldsymbol{\gamma})$-set if and only if

$$(-1)^{\boldsymbol{\sigma}} \not\leq \boldsymbol{\gamma}, \text{ for all } \boldsymbol{\sigma} \in \Xi, \quad \text{(B.1)}$$

$$(-1)^{\boldsymbol{\sigma}} + (-1)^{\boldsymbol{\sigma}'} \not\leq \boldsymbol{\gamma}, \text{ for all } \boldsymbol{\sigma}, \boldsymbol{\sigma}' \in \Xi. \quad \text{(B.2)}$$

Proof: If $\Xi \subset B^h$ is an $m(\boldsymbol{\gamma})$-set, then (B.1) and (B.2) follow from the definition.

Below we consider to prove that Ξ is an $m(\boldsymbol{\gamma})$-set if (B.1) and (B.2) are valid. The proof of cases $1 \leq h \leq 2$ is obvious. If $\gamma_j = 0$ holds for only one index j, say $j = j_0$, then from (B.1), we see that $\sigma_{j_0} = 0$ for all $\boldsymbol{\sigma} \in \Xi$ and thus (4.20) holds, i.e. Ξ is an $m(\boldsymbol{\gamma})$-set. Now we assume that $3 \leq h \leq 5$ and $\gamma_j = 0$ holds for at least two indices j.

At first, we consider the case of $\gamma_j \leq 1$ for all j. Assume in the contrast that Ξ is not an $m(\boldsymbol{\gamma})$-set. Then there is a tuple $\boldsymbol{q}' \in Q_h^*$ with $\sum_{\boldsymbol{\sigma}\in\Xi} q'_{\boldsymbol{\sigma}} \geq 3$ such that $\boldsymbol{\lambda} \triangleq \sum_{\boldsymbol{\sigma}\in\Xi} q'_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \leq \boldsymbol{\gamma}$. If some component of $\boldsymbol{\lambda}$ is 1, then the remaining components of $\boldsymbol{\lambda}$ must be also odd integers and thus at least two of them are not greater than -1. Hence $\sum_{j=1}^h \lambda_j \leq 1$ by $3 \leq h \leq 5$, and thus there is a tuple $\boldsymbol{\sigma}' \in \Xi$ with $q'_{\boldsymbol{\sigma}'} > 0$ such that $\sum_{i=1}^{h}(-1)^{\sigma'_i} \leq 0$, i.e. $w_{\rm H}(\boldsymbol{\sigma}') \geq \lceil h/2 \rceil$. Let $\tau \triangleq h - w_{\rm H}(\boldsymbol{\sigma}')$. Clearly, $1 \leq \tau \leq 2$ with respect to $3 \leq h \leq 5$ and $\mathbf{1} \notin \Xi$.

If $\tau = 2$, we assume, without loss of the generality, that $\sigma'_1 = \sigma'_2 = 0$ and $\sigma'_j = 1$, for $j > 2$. Thus, from $(-1)^{\boldsymbol{\sigma}'} \not\leq \boldsymbol{\gamma}$, we know that $\gamma_1 + \gamma_2 \leq 1$. For any $\boldsymbol{\sigma} \in \Xi$, from $(-1)^{\boldsymbol{\sigma}} + (-1)^{\boldsymbol{\sigma}'} \not\leq \boldsymbol{\gamma}$, we have $(-1)^{\sigma_1} + (-1)^{\sigma_2} \geq 0$. Then, we have $2 = (-1)^{\sigma'_1} + (-1)^{\sigma'_2} \leq \sum_{\boldsymbol{\sigma} \in \Xi} q'_{\boldsymbol{\sigma}}((-1)^{\sigma_1} + (-1)^{\sigma_2}) = \lambda_1 + \lambda_2 \leq \gamma_1 + \gamma_2 \leq 1$, it is a contradiction. If $\tau = 1$, a contradiction can also be deduced in a similar way.

Now we consider the case $\gamma_j \geq 2$ holds for some j's. From (B.1) and (B.2), we see that

$$(-1)^{p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\sigma})} \not\leq p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\gamma}), \tag{B.3}$$

$$(-1)^{p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\sigma})} + (-1)^{p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\sigma}')} \not\leq p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\gamma}), \tag{B.4}$$

for any $p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\sigma}), p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\sigma}') \in p_{\zeta(\boldsymbol{\gamma})}[\Xi]$. By the above proof, we know $p_{\zeta(\boldsymbol{\gamma})}[\Xi]$ is an $m(p_{\zeta(\boldsymbol{\gamma})}(\boldsymbol{\gamma}))$-set. Hence, Ξ is an $m(\boldsymbol{\gamma})$-set. □

Proof of Lemma 4.4: 1. Assume in the contrast that there is an $m(\mathbf{0})$-set Ξ such that

$$\xi'(-1)^{\boldsymbol{\sigma}^*} + \sum_{\boldsymbol{\sigma} \in \Xi} \xi'_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \leq \mathbf{0}, \tag{B.5}$$

$$\xi''(-1)^{\overline{\boldsymbol{\sigma}^*}} + \sum_{\boldsymbol{\sigma} \in \Xi} \xi''_{\boldsymbol{\sigma}}(-1)^{\boldsymbol{\sigma}} \leq \mathbf{0}, \tag{B.6}$$

hold for some tuple $\boldsymbol{\sigma}^* \in B^h \backslash \Xi$ and some positive integers ξ', ξ'' and some non-negative integers $\xi'_{\boldsymbol{\sigma}}, \xi''_{\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} \in \Xi$. Clearly, we have

$$\sum_{\boldsymbol{\sigma} \in \Xi} \xi'_{\boldsymbol{\sigma}} > 0 \text{ and } \sum_{\boldsymbol{\sigma} \in \Xi} \xi''_{\boldsymbol{\sigma}} > 0, \tag{B.7}$$

and then

$$\sum_{\boldsymbol{\sigma} \in \Xi} (\xi'\xi''_{\boldsymbol{\sigma}} + \xi''\xi'_{\boldsymbol{\sigma}})(-1)^{\boldsymbol{\sigma}} \leq 0, \tag{B.8}$$

contradicts that Ξ is an $m(\mathbf{0})$-set.

2. Assume that $h \leq 5$, $\boldsymbol{\gamma} \in \Gamma_0^h$ and Ξ is an $m(\boldsymbol{\gamma})$-set.

At first, we consider the case of $\gamma_j \leq 1$ for all j. Since $\boldsymbol{\gamma} \in \Gamma_0^h$, we have $(-1)^{\boldsymbol{\gamma}} \not\leq \mathbf{0}$. For any $\boldsymbol{\sigma} \in \Xi$, from $(-1)^{\boldsymbol{\sigma}} - \boldsymbol{\gamma} \not\leq \mathbf{0}$ and $(-1)^{\boldsymbol{\gamma}} \geq -\boldsymbol{\gamma}$, we see

$$(-1)^{\boldsymbol{\sigma}} + (-1)^{\boldsymbol{\gamma}} \not\leq \mathbf{0}. \tag{B.9}$$

Thus by Lemma B.1 we know $\Xi \cup \{\gamma\}$ is an $m(\mathbf{0})$-set. Then, there is an $m(\mathbf{0})$-set Ξ' such that $\Xi \cup \{\gamma\} \subset \Xi'$. Since for any sequences $\sigma, \sigma' \in \Xi'$,

$$(-1)^{\sigma} \not\leq -(-1)^{\gamma}, \tag{B.10}$$

$$(-1)^{\sigma} + (-1)^{\sigma'} \not\leq -(-1)^{\gamma}, \tag{B.11}$$

from $-(-1)^{\gamma} \leq \gamma$ and Lemma B.1 we know Ξ' is an $m(\gamma)$-set which contains Ξ.

If $\gamma_j \geq 2$ holds for some j's, by Lemma B.1 we know $p_{\zeta(\gamma)}[\Xi]$ is an $m(p_{\zeta(\gamma)}(\gamma))$-set. Let Ξ'_r be an $M(p_{\zeta(\gamma)}(\gamma))$-set which contains $p_{\zeta(\gamma)}[\Xi]$. Then

$$\{\sigma \in B^h : p_{\zeta(\gamma)}(\sigma) \in \Xi'_r\} \tag{B.12}$$

is an $M(\gamma)$-set which contains Ξ. □

Remark: According to the following lemma, we see that Lemma B.1 can not be generalized to the cases $h \geq 6$ directly.

Lemma B.2: Assume $\sigma^{(1)}$, $\sigma^{(2)}$, $\sigma^{(3)}$ and $\sigma^{(4)}$ are 4 tuples in B^h satisfying

$$(-1)^{\sigma^{(i)}} + (-1)^{\sigma^{(j)}} \not\leq \mathbf{0}, \text{ for } i, j = 1, 2, 3, 4. \tag{B.13}$$

Then there are nonnegative integers ξ_1, ξ_2, ξ_3 and ξ_4 with $\xi_1 + \xi_2 + \xi_3 + \xi_4 > 0$ such that

$$\sum_{j=1}^{4} \xi_j (-1)^{\sigma^{(j)}} \leq \mathbf{0}, \tag{B.14}$$

if and only if $h \geq 6$, $\xi_1 = \xi_2 = \xi_3 = \xi_4$,

$$\sum_{j=1}^{4} (-1)^{\sigma_i^{(j)}} \leq 0, \text{ for } i = 1, 2, \ldots, h, \tag{B.15}$$

and there are 6 indices $i_1, i_2, \ldots, i_6$ such that

$$\{(\sigma_{i_j}^{(1)}, \sigma_{i_j}^{(2)}, \sigma_{i_j}^{(3)}, \sigma_{i_j}^{(4)}) : 1 \leq j \leq 6\} = \{\sigma \in B^4 : \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 = 2\}. \tag{B.16}$$

Proof: Assume that there are nonnegative integers ξ_1, ξ_2, ξ_3 and ξ_4 such that (B.14) holds. From $(-1)^{\sigma^{(1)}} + (-1)^{\sigma^{(2)}} \not\leq \mathbf{0}$, we know there is at least one index i_{12} such that $\sigma_{i_{12}}^{(1)} = \sigma_{i_{12}}^{(2)} = 0$. Hence from

$$\sum_{j=1}^{4} \xi_j (-1)^{\sigma_{i_{12}}^{(j)}} \leq 0, \tag{B.17}$$

we know $\xi_1+\xi_2 \leq \xi_3+\xi_4$. Similarly, we can get $\xi_1+\xi_3 \leq \xi_2+\xi_4$, $\xi_1+\xi_4 \leq \xi_2+\xi_3$, $\xi_2+\xi_3 \leq \xi_1+\xi_4$, $\xi_2+\xi_4 \leq \xi_1+\xi_3$ and $\xi_3+\xi_4 \leq \xi_1+\xi_2$. Hence $\xi_1 = \xi_2 = \xi_3 = \xi_4$ and (B.15) hold for any $1 \leq i \leq h$. Furthermore, we know that $\alpha_{i_{12}}^{(3)} = \alpha_{i_{12}}^{(4)} = 1$, and consequently we can conclude that there are 6 indices $i_1, i_2, \ldots, i_6$ such that (B.16) holds, and $h \geq 6$.

On the other hand, assume that (B.15) and (B.16) hold for some indices $i_1, i_2, \ldots, i_6$. It is obvious that (B.13) is valid and (B.14) holds for $\xi_1 = \xi_2 = \xi_3 = \xi_4 = 1$. □

C Proof of Theorem 4.2

At first, we prove the following two lemmas. The latter one is indeed equivalent to Theorem 4.2.

Lemma C.1: Assume that $1 \leq h \leq 5$, $\gamma \in \Gamma_0^h$ and $\Xi \in \aleph_h$. Then Ξ is an $M(\gamma)$-set if and only if

$$p_{I(\Xi)}(\gamma) \in p_{I(\Xi)}[\Xi]. \tag{C.1}$$

Proof: Only-if-part: Let Ξ be an $M(\gamma)$-set. Then $p_{\zeta(\gamma)}[\Xi]$ is an $m(p_{\zeta(\gamma)}(\gamma))$-set by Lemma B.1. Since $|\Xi| = 2^{h-1}$ and

$$\Xi \subset \Xi' = \{\sigma \in B^h : p_{\zeta(\gamma)}(\sigma) \in p_{\zeta(\gamma)}[\Xi]\}, \tag{C.2}$$

we know $p_{\zeta(\gamma)}[\Xi]$ is an $M(p_{\zeta(\gamma)}(\gamma))$-set, and $\Xi = \Xi'$, i.e. (4.23) holds for $I = \zeta(\gamma)$ and then

$$I(\Xi) \subseteq \zeta(\gamma). \tag{C.3}$$

From $(-1)^{\overline{p_{\zeta(\gamma)}(\gamma)}} \leq p_{\zeta(\gamma)}(\gamma)$, we know $\overline{p_{\zeta(\gamma)}(\gamma)} \notin p_{\zeta(\gamma)}[\Xi]$, and then

$$p_{\zeta(\gamma)}(\gamma) \in p_{\zeta(\gamma)}[\Xi]. \tag{C.4}$$

The equality (C.1) follows from (C.3) and (C.4).

If-part: Assume that (C.1) is valid. Since Ξ is $m(\mathbf{0})$-set, we know $p_{I(\Xi)}[\Xi]$ is an $m(\mathbf{0})$-set, and thus by (C.1) and $-(-1)^{p_{I(\Xi)}(\gamma)} \leq p_{I(\Xi)}(\gamma)$, we know $p_{I(\Xi)}[\Xi]$ is an $m(p_{I(\Xi)}(\gamma))$-set. Then Ξ is $m(\gamma)$-set, and thus Ξ is an $M(\gamma)$-set because of $|\Xi| = 2^{h-1}$. □

Lemma C.2: Assume that $1 \le h \le 5$ and $\boldsymbol{q} \in Q_h$. Then $\boldsymbol{q}$ belongs to $Q_h^{\min}$ if and only if $S(\boldsymbol{q})$ is a subset of some $M(\mathbf{0})$-set Ξ and

$$p_{I(\Xi)}(\gamma(\boldsymbol{q})) = p_{I(\Xi)}(\boldsymbol{\sigma}(\Xi)), \tag{C.5}$$

$$p_{[1,h]\setminus I(\Xi)}(\gamma(\boldsymbol{q})) \ge p_{[1,h]\setminus I(\Xi)}(\boldsymbol{\sigma}(\Xi)). \tag{C.6}$$

Proof: Only-if-part: Assume that $\boldsymbol{q}$ belongs to $Q_h^{\min}$. By Lemma 4.2, we know $(-1)^{\boldsymbol{\sigma}} \not\le \gamma(\boldsymbol{q})$ and $(-1)^{\boldsymbol{\sigma}} + (-1)^{\boldsymbol{\sigma}'} \not\le \gamma(\boldsymbol{q})$ for all $\boldsymbol{\sigma}, \boldsymbol{\sigma}' \in S(\boldsymbol{q})$. Thus, by Lemma B.1, we know $S(\boldsymbol{q})$ is an $m(\gamma(\boldsymbol{q}))$-set. Hence, by Lemma 4.4, we know $S(\boldsymbol{q})$ is a subset of some $M(\gamma(\boldsymbol{q}))$-set Ξ. Then by Lemma C.1, we have

$$p_{I(\Xi)}(\gamma(\boldsymbol{q})) \in p_{I(\Xi)}[\Xi]. \tag{C.7}$$

Since $\gamma(\boldsymbol{q}) + \boldsymbol{\delta} \in \Lambda_0^h \cup \Lambda_1^h$, there is a tuple $\boldsymbol{\sigma} \in B^h$ satisfying (4.22) and

$$p_{I(\Xi)}(\boldsymbol{\sigma}) = p_{I(\Xi)}(\gamma(\boldsymbol{q})) \tag{C.8}$$

Clearly, $\gamma(\boldsymbol{q}) \ge \boldsymbol{\sigma}$. From (C.7) and (C.8), we see that $\boldsymbol{\sigma}$ belongs to Ξ and thus $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\Xi)$. Hence, (C.5) and (C.6) are valid.

If-part: Assume that $S(\boldsymbol{q})$ is a subset of some $M(\mathbf{0})$-set Ξ, (C.5) and (C.6) are valid. By Lemma C.1, we know that Ξ is an $M(\gamma(\boldsymbol{q}))$-set. Hence, $S(\boldsymbol{q})$ is an $m(\gamma(\boldsymbol{q}))$-set. Thus $\boldsymbol{q}$ belongs to $Q_h^{\min}$ by Lemma 4.2. □

Proof of Theorem 4.2:

It is obvious that Theorem 4.2 is equivalent to Lemma C.2. □

D Proof of Theorem 4.3

Proof of Theorem 4.3: The proof for the case $1 \le h \le 3$ is very simple. Below we only prove the cases of $h = 4$ and $h = 5$.

We first consider the case of $h = 4$.

If $\Xi \in \{C_i, D_i, E_i\}_{i=1}^4$, $\boldsymbol{\sigma}, \boldsymbol{\sigma}' \in \Xi$, then there is at least one i such that $\sigma_i = \sigma_i' = 0$. Hence $(-1)^{\boldsymbol{\sigma}} \not\le \mathbf{0}$ and $(-1)^{\boldsymbol{\sigma}} + (-1)^{\boldsymbol{\sigma}'} \not\le \mathbf{0}$ for any $\boldsymbol{\sigma}, \boldsymbol{\sigma}' \in \Xi$, and by Lemma B.1, we know Ξ is a $M(\mathbf{0})$-set.

Now we suppose Ξ is a $M(\mathbf{0})$-set. If there is an i_0 such that $\sigma_{i_0} = 0$ for all $\boldsymbol{\sigma} \in \Xi$, then we have $\Xi = C_{i_0}$. If there is a tuple $\boldsymbol{\sigma}' \in \Xi$ such that $w_{\mathrm{H}}(\boldsymbol{\sigma}') = 3$, then we also have $\Xi = C_{i_0}$ for some index i_0. Now we assume $\sigma_i^{(i)} = 1$ for four sequences $\sigma^{(i)} \in \Xi$, $i = 1, 2, 3, 4$, and $w_{\mathrm{H}}(\sigma) \le 2$ for all $\sigma \in \Xi$. Let Φ denote the set of the sequences $\sigma \in \Xi$ with $w_{\mathrm{H}}(\sigma) = 2$. By $|\Xi| = 2^3$, we know $|\Phi| \ge 3$. We assume, without loss of the generality, that $1100, 1010 \in \Phi$. Clearly, $\Phi \setminus \{1100, 1010\} \subseteq \{0110, 1001\}$. Then, from $(-1)^{0110} + (-1)^{1001} = \mathbf{0}$, we know $\Xi \in \{E_4, D_1\}$.

Next we consider the case of $h = 5$.

Similar to the case $h = 4$, if Ξ is one of the sets given in (4.33) to (4.38), then we can also show easily that Ξ is a $M(\mathbf{0})$-set.

Now we suppose that Ξ is a $M(\mathbf{0})$-sets. If there is a $\sigma \in \Xi$ satisfies $w_{\mathrm{H}}(\sigma) = 4$, then $\Xi = G_{i_0}$ holds for some index i_0. Next we assume that $w_{\mathrm{H}}(\sigma) \le 3$ for all $\sigma \in \Xi$. Let $\mathfrak{R}$ denote the set of the sequences $\sigma \in \Xi$ with $w_{\mathrm{H}}(\sigma) = 3$.

If $\mathfrak{R}$ is a empty set, then $\Xi = F$.

If $\mathfrak{R} = \{\sigma\}$, then $\overline{\sigma} \notin \Xi$ and $\Xi = K_{\overline{\sigma}, \overline{\sigma}}$.

If $\mathfrak{R} = \{\sigma', \sigma''\}$, then $\{\overline{\sigma'}, \overline{\sigma''}\} \cap \Xi = \phi$, and $\sigma'_i = \sigma''_i = 0$ for some i. Hence we have $\Xi = K_{\overline{\sigma'}, \overline{\sigma''}}$.

If $|\mathfrak{R}| \ge 3$, without loss of the generality, we assume that 11100, 11010 $\in \mathfrak{R}$. Then $\mathfrak{R} \setminus \{11100, 11010\} \subseteq \{11001, 01110, 10110\}$. Hence, $\Xi \in \{I_{345}, J^5_{134}, J^5_{123}\}$ if $|\mathfrak{R}| = 3$, and from $(-1)^{11001} + (-1)^{01110} \le \mathbf{0}$ and $(-1)^{11001} + (-1)^{10110} \le \mathbf{0}$, we know $\Xi = H_5$ if $|\mathfrak{R}| \ge 4$. □

E Proof of Lemma 4.6

Proof of Lemma 4.6: We only prove for the case $\min\{n_{010}, n_{001}\} > 0$. The proof for the case $\min\{n_{010}, n_{001}\} = 0$ is simpler and can be obtained in a similar way. Let

$$n_{010}(k) \triangleq |\mathcal{D}_{010}^{(\lfloor(\delta_1-\delta_2)/2\rfloor-k)} \setminus \{\infty\}|, \tag{E.1}$$

$$n_{001}(k) \triangleq |\mathcal{D}_{001}^{(\lfloor(\delta_1-\delta_3)/2\rfloor-k)} \setminus \{\infty\}|. \tag{E.2}$$

1. For k with $0 \le k \le k_0$, we see that ∞ is not in $X(k)$ if and only if

$$\psi(k) \triangleq n_{000} + n_{010}(k) + n_{001}(k) - (\delta_1 - k) \ge 0. \tag{E.3}$$

Let k_1^{l} and k_1^{r} denote the smallest and the largest k in $[0, k_0]$ for which (E.3) holds respectively. Clearly, in the interval $[0, k_0 - 1]$, $\psi(k) - \psi(k+1)$ is nondecreasing and satisfies $|\psi(k) - \psi(k+1)| \leq 1$. If $\infty \notin X(k)$ and $\psi(k) - \psi(k+1) \leq 0$, we see $\infty \notin X(k+1)$ easily. If there is an integer k_2 with $k_1^{\mathrm{l}} \leq k_2 < k_1^{\mathrm{r}}$ such that $\infty \notin X(k_2)$ and $\psi(k_2) - \psi(k_2+1) = 1$, then, for any k with $k_2 < k \leq k_1^{\mathrm{r}}$, from $\psi(k_2) = \psi(k_1^{\mathrm{r}}) + (k_1^{\mathrm{r}} - k_2) \geq k_1^{\mathrm{r}} - k_2$ we see $\psi(k) = \psi(k_2) - (k - k_2) \geq k_1^{\mathrm{r}} - k \geq 0$, this implies $\infty \notin X(k)$. Hence, $\infty \notin X(k)$ for all k with $k_1^{\mathrm{l}} \leq k \leq k_1^{\mathrm{r}}$.

2. Clearly, for k with $k_{\mathrm{l}} \leq k < k_{\mathrm{r}}$ we have

$$\min\{n_{010}(k), n_{001}(k)\} > 0. \tag{E.4}$$

We say $\boldsymbol{q}^k$ is a-type, or b-type, or c-type, or d-type if $q_{010}^k < \delta_{12} - k$ and $q_{001}^k < \delta_{13} - k$, or $q_{010}^k < \delta_{12} - k$ and $q_{001}^k = \delta_{13} - k$, or $q_{010}^k = \delta_{12} - k$ and $q_{001}^k < \delta_{13} - k$, or $q_{010}^k = \delta_{12} - k$ and $q_{001}^k = \delta_{13} - k$, respectively. Write $i(k) = \max\{i : i \in X(k)\}$, $i_{010}(k) = \max\{i : i \in X(k) \cap \mathcal{D}_{010}\}$, $i_{001}(k) = \max\{i : i \in X(k) \cap \mathcal{D}_{001}\}$ and $i_{000}(k) = \min\{i : i \in \mathcal{D}_{000} \backslash X(k)\}$.

If $\boldsymbol{q}^k$ is a-type, we see $X(k+1) = X(k) \backslash \{i(k)\}$ and $\mu(k) = |r_{i(k)}|$.

If $\boldsymbol{q}^k$ is b-type, we see $X(k) \cap \mathcal{D}_{001} \neq \phi$, $X(k+1) = X(k) \backslash \{i_{001}(k)\}$ and $\mu(k) = |r_{i_{001}(k)}|$. Furthermore, if $k+1 < k_{\mathrm{r}}$, $\boldsymbol{q}^{k+1}$ must be b-type or d-type.

If $\boldsymbol{q}^k$ is c-type, we see $X(k) \cap \mathcal{D}_{010} \neq \phi$, $X(k+1) = X(k) \backslash \{i_{010}(k)\}$ and $\mu(k) = |r_{i_{010}(k)}|$. Furthermore, if $k+1 < k_{\mathrm{r}}$, $\boldsymbol{q}^{k+1}$ must be c-type or d-type.

If $\boldsymbol{q}^k$ is d-type, we see that $X(k) \cap \mathcal{D}_{010}$, $X(k) \cap \mathcal{D}_{001}$ and $\mathcal{D}_{000} \backslash X(k)$ are non-empty sets, $X(k+1) = (X(k) \backslash \{i_{010}(k), i_{001}(k)\}) \cup \{i_{000}(k)\}$ and $\mu(k) = |r_{i_{010}(k)}| + |r_{i_{001}(k)}| - |r_{i_{000}(k)}|$. Clearly, we have $|r_{i_{000}(k)}| \geq \max\{|r_{i_{010}(k)}|, |r_{i_{001}(k)}|\}$ and thus

$$\min\{|r_{i_{010}(k)}|, |r_{i_{001}(k)}|\} \geq \mu(k). \tag{E.5}$$

Furthermore, if $k+1 < k_{\mathrm{r}}$, $\boldsymbol{q}^{k+1}$ must be d-type.

Clearly, we see $i_{010}(k) \geq i_{010}(k+1)$ and $i_{001}(k) \geq i_{001}(k+1)$. Thus, with respect to (E.5), we can deduce easily that $\mu(k+1) \leq \mu(k)$ if $k+1 < k_{\mathrm{r}}$. □

F Proof of Lemma 4.8

Proof of Lemma 4.8: If $R^{d,k+1}$ is empty, then $\xi(\boldsymbol{q}^{d,k})$ is also empty by its definition. We will prove that there exists an $R^{d,k+1}$-optimum in the set $\xi(\boldsymbol{q}^{d,k})$

if $R^{d,k+1}$ is not empty. Assume $R^{d,k+1} \neq \phi$ and $\boldsymbol{q}^{d,k+1}$ is an $R^{d,k+1}$-optimum. Let $\mathcal{A}_1$ denote the set of pairs

$$(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in \{(0000, 0001), (0100, 0101), (0010, 0011), (0110, 0111)\} \tag{F.1}$$

which satisfy

$$q_{\boldsymbol{\sigma}'}^{d,k+1} > q_{\boldsymbol{\sigma}'}^{d,k} \text{ and } q_{\boldsymbol{\sigma}''}^{d,k+1} < q_{\boldsymbol{\sigma}''}^{d,k}, \tag{F.2}$$

If $\mathcal{A}_1 \neq \phi$, then any pair $(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in \mathcal{A}_1$ must satisfy

$$\varphi(\boldsymbol{q}^{d,k}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in R^{d,k+1} \text{ and } \varphi(\boldsymbol{q}^{d,k+1}, \boldsymbol{\sigma}'', \boldsymbol{\sigma}') \in R^{d,k}. \tag{F.3}$$

Let $l_{\boldsymbol{\sigma}'}^{d,k+1}$ and $l_{\boldsymbol{\sigma}'}^{d,k}$ denote the largest integers in $\mathcal{D}_{\boldsymbol{\sigma}'}^{(q_{\boldsymbol{\sigma}'}^{d,k+1})}$ and $\mathcal{D}_{\boldsymbol{\sigma}''}^{(q_{\boldsymbol{\sigma}''}^{d,k})}$ respectively. Let $s_{\boldsymbol{\sigma}'}^{d,k}$ and $s_{\boldsymbol{\sigma}''}^{d,k+1}$ denote the smallest integers in $\mathcal{D}_{\boldsymbol{\sigma}'} \backslash \mathcal{D}_{\boldsymbol{\sigma}'}^{(q_{\boldsymbol{\sigma}'}^{d,k})}$ and $\mathcal{D}_{\boldsymbol{\sigma}''} \backslash \mathcal{D}_{\boldsymbol{\sigma}''}^{(q_{\boldsymbol{\sigma}''}^{d,k+1})}$ respectively. Clearly, we have $l_{\boldsymbol{\sigma}'}^{d,k+1} \geq s_{\boldsymbol{\sigma}'}^{d,k}$ and $s_{\boldsymbol{\sigma}''}^{d,k+1} \leq l_{\boldsymbol{\sigma}''}^{d,k}$ and thus

$$\begin{aligned} L'(\boldsymbol{q}^{d,k+1}) &= L'(\varphi(\boldsymbol{q}^{d,k+1}, \boldsymbol{\sigma}'', \boldsymbol{\sigma}')) + |\boldsymbol{r}_{l_{\boldsymbol{\sigma}'}^{d,k+1}}| - |\boldsymbol{r}_{s_{\boldsymbol{\sigma}''}^{d,k+1}}| \\ &\geq L'(\boldsymbol{q}^{d,k}) + |\boldsymbol{r}_{s_{\boldsymbol{\sigma}'}^{d,k}}| - |\boldsymbol{r}_{l_{\boldsymbol{\sigma}''}^{d,k}}| = L'(\varphi(\boldsymbol{q}^{d,k}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')), \end{aligned} \tag{F.4}$$

this implies that $\varphi(\boldsymbol{q}^{d,k}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')$ is a $R^{d,k+1}$-optimum.

If $\mathcal{A}_1 = \phi$, then from the first equality in (4.60) and $(0000, 0001) \notin \mathcal{A}_1$ we see that

$$q_{0000}^{d,k+1} \leq q_{0000}^{d,k} \text{ and } q_{0001}^{d,k+1} \geq q_{0001}^{d,k}, \tag{F.5}$$

and thus from (4.67) and the second equality in (4.60) we see that there must be at least one pair

$$(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') \in \{0100, 0010, 0110\} \times \{0101, 0011, 0111\}. \tag{F.6}$$

such that (F.2) holds. We conclude that this pair $(\boldsymbol{\sigma}', \boldsymbol{\sigma}'')$ must satisfy (F.3) too and thus $\varphi(\boldsymbol{q}^{d,k}, \boldsymbol{\sigma}', \boldsymbol{\sigma}'')$ is a $R^{d,k+1}$-optimum. Indeed, for clarify, say $(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') = (0100, 0011)$ satisfies (F.2), i.e.

$$q_{0100}^{d,k+1} > q_{0100}^{d,k} \text{ and } q_{0011}^{d,k+1} < q_{0011}^{d,k}, \tag{F.7}$$

since both of $(0100, 0101)$ and $(0010, 0011)$ are not in $\mathcal{A}_1$, we see that

$$q_{0101}^{d,k+1} \geq q_{0101}^{d,k} \text{ and } q_{0010}^{d,k+1} \leq q_{0010}^{d,k}. \tag{F.8}$$

From (F.7) and (F.8), we see that (F.3) holds for $(\boldsymbol{\sigma}', \boldsymbol{\sigma}'') = (0100, 0011)$. □

G Proof of Lemma 4.10

To prove Lemma 4.10, we first introduce some more definitions and another lemma. For $\pi \in \Omega(w)$, let $\mathcal{G}(\pi)$ denote the set of pairs $\pi' \in \Upsilon$ with $L_w(\pi) \leq L_w(\pi + \pi')$. For any pair $\pi' = (x_1', x_2', x_3') \in \Upsilon$, let $p(\pi')$ denote 7-tuple

$$(x_1',\ x_2',\ x_3',\ x_1' + x_2',\ x_1' + x_3',\ x_2' + x_3',\ x_1' + x_2' + x_3'), \tag{G.1}$$

and let $\Psi(\pi')$ denote the set of pairs $\pi'' = (x_1'', x_2'', x_3'') \in \Upsilon$ which satisfy

$$p(\pi')_j \cdot p(\pi'')_j \geq 0, \text{ for } j = 1, 2, \ldots, 7. \tag{G.2}$$

We say set $\Upsilon' \subseteq \Upsilon$ **coherent** if $\Upsilon' \subseteq \Psi(\pi')$ for each $\pi' \in \Upsilon'$.

Lemma G.1: 1). If $\pi' \in \mathcal{G}(\pi) \cap \Psi(\pi'')$, we must have $\pi' \in \mathcal{G}(\pi + \pi'')$.
2). If $\{\pi_1, \pi_2, \ldots, \pi_k\} \subseteq \mathcal{G}(\pi)$ is coherent and $t_1, t_2, \ldots, t_k$ are nonnegative integers, we must have $L_w(\pi) \leq L_w(\pi + \sum_{i=1}^k t_i \pi_i)$.
3). A pair $\pi \in \Omega(w)$ is a $\Omega(w)$-pair if and only if $\Upsilon \subseteq \mathcal{G}(\pi)$.
4). A pair $\pi \in \Omega(w, x)$ is a $\Omega(w, x)$-pair if and only if

$$\Upsilon_1 \triangleq \{(1,0,0), (-1,0,0), (0,1,0), (0,-1,0), (1,-1,0), (-1,1,0)\} \subseteq \mathcal{G}(\pi).$$

Proof. We only prove the result 3) here. The results 1) and 2) can be proved easily from the definitions, and the result 4) can be proved with a similar method of proving the result 3).

If $\pi \in \Omega(w)$ is a $\Omega(w)$-pair, we see easily that $\Upsilon \subseteq \mathcal{G}(\pi)$ holds. Now we assume that $\pi \in \Omega(w)$ satisfies $\Upsilon \subseteq \mathcal{G}(\pi)$. We will prove $L_w(\pi') \geq L_w(\pi)$ for any $\pi' \in \Omega(w)$ and thus π is a $\Omega(w)$-pair. Let π' be an arbitrary pair in $\Omega(w)$ and write $\pi^* = (x_1^*, x_2^*, x_3^*)$ for $\pi' - \pi$. If all of x_1^*, x_2^*, x_3^* are nonnegative or nonpositive, from $\{(1,0,0), (0,1,0), (0,0,1)\}$ and $\{(-1,0,0), (0,-1,0), (0,0,-1)\}$ are coherent, we know $L_w(\pi') \geq L_w(\pi)$ easily. For the other cases, without loss of generality, we assume that $x_1^* \geq x_2^* \geq 0, x_3^* < 0$.

If $-x_3^* \leq x_2^*$, from $\pi^* = (x_1^* + x_3^*)(1,0,0) + (x_2^* + x_3^*)(0,1,0) + (-x_3^*)(1,1,-1)$ and $\{(1,0,0), (0,1,0), (1,1,-1)\}$ is coherent, we know $L_w(\pi') \geq L_w(\pi)$.

If $x_2^* < -x_3^* \leq x_1^*$, from $\pi^* = (x_1^* + x_3^*)\ (1,0,0) + (-x_2^* - x_3^*)(1,0,-1) + x_2^*(1,1,-1)$ and $\{(1,0,0), (1,0,-1), (1,1,-1)\}$ is coherent, we know $L_w(\pi') \geq L_w(\pi)$.

If $x_1^* < -x_3^* < x_1^* + x_2^*$, from $\pi^* = (-x_2^* - x_3^*)(1,0,-1) + (-x_1^* - x_3^*)(0,1,-1) + (x_1^* + x_2^* + x_3^*)(1,1,-1)$ and $\{(1,0,-1),(0,1,-1),(1,1,-1)\}$ is coherent, we know $L_w(\pi') \geq L_w(\pi)$.

If $-x_3^* \geq x_1^* + x_2^*$, from $\pi^* = (-x_1^* - x_2^* - x_3^*)(0,0,-1) + x_1^*(1,0,-1) + x_2^*(0,1,-1)$ and $\{(0,0,-1),\ (1,0,-1),(0,1,-1)\}$ is coherent, we know $L_w(\pi') \geq L_w(\pi)$. □

Proof of Lemma 4.10: Let π_1 denote the output pair of Step 1. Without loss of generality, we assume that $\{(1,0,0),(-1,0,0),(0,-1,0)\} \subseteq \mathcal{G}(\pi_1)$. We consider to grow π_1 in the $(0,1,0)$-direction till we can not do anymore, and let π_1' denote the result pair. Clearly, we have $\{(0,1,0),(0,-1,0)\} \subseteq \mathcal{G}(\pi_1')$. From $(1,0,0) \in \Psi(0,1,0) \cap \mathcal{G}(\pi_1)$ and $\pi_1' = \pi_1 + k(0,1,0)$ for some nonnegative integer k, we see $(1,0,0) \in \mathcal{G}(\pi_1')$. Next we consider to grow π_1' in the $(-1,0,0)$-direction till we can not do anymore, then result pair π_1'' must satisfy $\{(1,0,0),(-1,0,0),(0,-1,0)\} \subseteq \mathcal{G}(\pi_1'')$. And so on, we see that $\{(1,0,0),(0,1,0),\ (-1,0,0),(0,-1,0)\} \subseteq \mathcal{G}(\pi_2)$ for the output pair π_2 of Step 2.

Clearly, we have either $(1,-1,0) \in \mathcal{G}(\pi_2)$ or $(-1,1,0) \in \mathcal{G}(\pi_2)$. Furthermore, if π_2 is grown from $\pi_2 - (0,1,0)$, from $(1,0,0) \in \mathcal{G}(\pi_2 - (0,1,0))$ we see $L_w(\pi_2 + (1,-1,0)) = L_w((\pi_2 - (0,1,0)) + (1,0,0)) \geq L_w(\pi_2 - (0,1,0)) \geq L_w(\pi_2)$ and thus $(1,-1,0) \in \mathcal{G}(\pi_2)$. Similarly, we can show that $(1,-1,0) \in \mathcal{G}(\pi_2)$ if π_2 is grown from $\pi_2 - (-1,0,0)$, and $(-1,1,0) \in \mathcal{G}(\pi_2)$ if π_2 is grown from $\pi_2 - (1,0,0)$ or $\pi_2 - (0,-1,0)$. Without loss of generality, we assume that $\Upsilon_0 \triangleq \{(1,0,0),(0,1,0),(-1,0,0),(0,-1,0),(1,-1,0)\} \subseteq \mathcal{G}(\pi_2)$ and $(-1,1,0) \notin \mathcal{G}(\pi_2)$. According to $L_w(\pi_2 + (-1,1,0)) < L_w(\pi_2) \leq \min\{L_w(\pi_2 + (0,1,0)), L_w(\pi_2 + (-1,0,0))\}$, we know $\{(1,-1,0),(1,0,0),(0,-1,0)\} \subseteq \mathcal{G}(\pi_2 + (-1,1,0))$. From $\{(-1,0,0),(0,1,0)\} \subseteq \mathcal{G}(\pi_2) \cap \Psi(-1,1,0)$, we see also $\{(-1,0,0),(0,1,0)\} \subseteq \mathcal{G}(\pi_2 + (-1,1,0))$. Thus, we have $\Upsilon_0 \subseteq \mathcal{G}(\pi_2 + (-1,1,0))$ and further $\Upsilon_0 \subseteq \mathcal{G}(\tau(w,x))$ for the output pair $\tau(w,x)$ of the procedure. Since $(-1,1,0)$ is also in $\mathcal{G}(\tau(w,x))$, from 3) of Lemma G.1 we see that $\tau(w,x)$ is a $\Omega(w,x)$-pair.

According to the above argument, we see easily that the growth route must be one of the four classes shown in Fig. 4.1. □

H Proof of Lemma 4.12

Proof of Lemma 4.12: We only prove the result 1, the result 2 can be proved by a similar method and the result 3 is a direct corollary of the results 3) and 4) of Lemma G.1. Let π_0 denote the pair in Υ^* which satisfies

$$L_w(\tau(w,x) + (0,0,1) + \pi_0) = \min_{\pi \in \Upsilon^*} L_w(\tau(w,x) + (0,0,1) + \pi). \tag{H.1}$$

Below we prove that $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x)+(0,0,1)+\pi_0)$, and thus $\tau(w,x)+(0,0,1)+\pi_0$ is a $\Omega(w, x+1)$-pair by 4) of Lemma G.1.

If $\pi_0 = (0,0,0)$, by (H.1) we know $\{(-1,0,0),(0,-1,0),(1,-1,0),(-1,1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (0,0,1))$. Since $\{(1,0,0),(0,1,0)\} \subseteq \Psi(0,0,1) \cap \mathcal{G}(\tau(w,x))$, we know $\{(1,0,0),(0,1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (0,0,1))$ also holds, and thus $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x) + (0,0,1) + \pi_0)$.

If $\pi_0 = (-1,0,0)$, then by (H.1) we know $\{(1,0,0),(0,1,0),(0,-1,0),(1,-1,0)\} \subseteq \mathcal{G}(\tau(w,x)+(-1,0,1))$. Since $\{(-1,0,0),(-1,1,0)\} \subseteq \mathcal{G}(\tau(w,x))\cap\Psi(-1,0,1)$, we see $\{(-1,0,0),(-1,1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (-1,0,1))$ also holds, and thus $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x) + (0,0,1) + \pi_0)$.

If $\pi_0 = (0,-1,0)$, similar to the above case, we can show $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x) + (0,0,1) + \pi_0)$.

If $\pi_0 = (-1,-1,0)$, then by (H.1) we know $\{(1,0,0),(0,1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (-1,-1,1))$. Since $\{(-1,0,0),(0,-1,0)\} \subseteq \Psi(-1,-1,1) \cap \mathcal{G}(\tau(w,x))$, we know $\{(-1,0,0),(0,-1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (-1,-1,1))$ also holds. On the other hand, from $(0,-1,0) \in \Psi(0,-1,1)\cap\mathcal{G}(\tau(w,x))$ and $(-1,0,0) \in \Psi(-1,0,1)\cap\mathcal{G}(\tau(w,x))$, we know $(0,-1,0) \in \mathcal{G}(\tau(w,x) + (0,-1,1))$ and $(-1,0,0) \in \mathcal{G}(\tau(w,x) + (-1,0,1))$. Then we see $L_w(\tau(w,x) + (0,-2,1)) \geq L_w(\tau(w,x) + (0,-1,1)) \geq L_w(\tau(w,x)+(-1,-1,1))$ and $L_w(\tau(w,x)+(-2,0,1)) \geq L_w(\tau(w,x)+(-1,0,1)) \geq L_w(\tau(w,x)+(-1,-1,1))$, hence $\{(1,-1,0),(-1,1,0)\} \subseteq \mathcal{G}(\tau(w,x)+(-1,-1,1))$. Thus $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x) + (0,0,1) + \pi_0)$.

If $\pi_0 = (1,-1,0)$, then by (H.1) we know $\{(-1,0,0),(-1,1,0)\} \subseteq \mathcal{G}(\tau(w,x)+(1,-1,1))$. Since $\{(1,0,0),(1,-1,0)\} \subseteq \Psi(1,-1,1)\cap\mathcal{G}(\tau(w,x))$, we know $\{(1,0,0),(1,-1,0)\} \subseteq \mathcal{G}(\tau(w,x) + (1,-1,1))$ also holds. On the other hand, from $(1,-1,0) \in \Psi(0,-1,1)\cap\mathcal{G}(\tau(w,x))$ and $(1,0,0) \in \Psi(0,0,1)\cap\mathcal{G}(\tau(w,x))$, we know $(1,-1,0) \in \mathcal{G}(\tau(w,x) + (0,-1,1))$ and $(1,0,0) \in \mathcal{G}(\tau(w,x) + (0,0,1))$. Then we see $L_w(\tau(w,x) + (1,-2,1)) \geq L_w(\tau(w,x) + (0,-1,1)) \geq L_w(\tau(w,x) + (1,-1,1))$

and $L_w(\tau(w,x)+(1,0,1)) \geq L_w(\tau(w,x)+(0,0,1)) \geq L_w(\tau(w,x)+(1,-1,1))$, hence $\{(0,-1,0),(0,1,0)\} \in \mathcal{G}(\tau(w,x)+(1,-1,1))$. And thus $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x)+(0,0,1)+\pi_0)$.

If $\pi_0 = (-1,1,0)$, similar to the above case, we can get $\Upsilon_1 \subseteq \mathcal{G}(\tau(w,x)+(0,0,1)+\pi_0)$. □