Successively Solvable Shift-Add Systems — a Graphical Characterization

Xiaopeng Cheng, Ximing Fu, Yuanxin Guo, Kenneth W. Shum and Shenghao Yang

***Abstract*—In order to reduce computational complexity in data encoding, one can use bitwise shifts and logical XOR operations instead of more costly calculations, and apply a fast decoding method called zigzag decoding. Existing works on zigzag decoding usually design special generator matrices that enable certain zigzag solving algorithms. In this paper, we study this class of fast decoding methods holistically. The shift operations are represented by a shift matrix, whose entries are integers or a special infinity symbol. A negative entry signifies that some symbols are truncated, and an infinite symbol means that the corresponding input sequence is not involved in the encoding process. Two notions of solvability, called successive solvability and zigzag solvability, are formulated. The former is employed in most of the existing works on zigzag decoding, and is a special case of the latter one. We prove in this paper that these two notions of solvability are equivalent when the shift matrix have no negative entries. An equivalent condition for a successively solvable shift-XOR system is derived in terms of a directed graph, when the shift matrix has only finite entries. This characterization result reveals the structure and the interconnections between the problem instances.**

I. Introduction

Due to the advantage of low encoding and decoding computation costs, (non-cyclic) shift and XOR operations have been extensively studied in the past several years for constructing storage codes [1]–[6], regenerating codes [7]–[10], fountain codes [11]–[14] and network codes [15]. The encoding of shift-XOR codes involves only XOR calculations. Using n message sequences, each consisting of L bits, the encoding of a (non-systematic) coded sequence requires at most $(n-1)L$ XOR operations.

The decoding problem of these shift-XOR codes is equivalent to solving an $n \times n$ system of equations with binary polynomials as the coefficients. When the coefficients satisfy certain conditions, we can apply a low-complexity zigzag decoding in solving the system of equations. The procedure in zigzag decoding is similar to successive interference cancellation as in the decoding of fountain codes [16], [17], network codes [18], [19], and LDPC codes for erasure channels [20]. Existing works have characterized several classes of codes that are zigzag decodable. When the generator matrix satisfies the refined increasing difference (RID) property, a zigzag decoding algorithm can be applied [1], [4], [10], which consumes the same number of XOR operations as in the encoding the n coded sequences. Another class of generator matrices that are zigzag solvable has a circulant structure [5]. Greedy algorithms have been developed to determine whether a given shift-XOR system is zigzag solvable [1], [19]. As far as we know, however, a systematic treatment of zigzag solvability is not available in literature.

In this paper, we study $n \times n$ shift-XOR systems where the shift operations can be represented by a *shift matrix* with integers, and possibly an ∞ symbol, as the entries. Here a negative entry means truncation of the corresponding sequence and ∞ means the corresponding sequence is not involved. Similar representation of shift-XOR systems has been used in literature, e.g., [19]. In Sec. II, we present the formal definitions of *zigzag solvability* and *successive solvability*. The latter is a special case of the former, by further requiring that the symbols in a sequence are solved in the ascending order of the symbol indices. The definition of successive solvability is general enough to include RID and circulant generator matrices studied in [1], [2], [4], [5]. Moreover, the successive solvability and zigzag solvability are proved to be equivalent when the shift matrix does not have negative entries.

Our main result, presented in Sec. IV, is a necessary and sufficient condition for a shift-XOR system to be successively solvable when the shift matrix only has finite entries. Our characterization is based on a graphical description of shift matrices with finite entries. If one step of successive cancellation is applied to a shift-XOR system, a new system is obtained. This relation connects one shift matrix to another one, and a directed graph is hence created. We then show that a shift matrix with only finite entries is successively solvable if and only if it is in or connects to a strongly connected component formed by cycles of shift matrices. See details in Sec. III and IV. In the concluding remarks, we discuss how to extend this characterization to shift matrices with ∞ entries.

II. Shift-Add Systems and Solvabilities

We model a data symbol as a value in a finite abelian group $\mathcal{A}$ with binary operation $+$ and identity element 0. One example of $\mathcal{A}$ is the set $\{0, 1\}$ with exclusive OR as the binary operation. We denote sequences with entries from $\mathcal{A}$ by lowercase letters in boldface, e.g., $\mathbf{s}$, where the i-th entry is denoted by $\mathbf{s}[i]$, for $i = 0, 1, 2, \ldots$ We denote the set of nonnegative integers by $\mathbb{N}_0$ and the set of integers by $\mathbb{Z}$. Formally speaking, an infinite sequence $\mathbf{s}$ is a function mapping from $\mathbb{N}_0$ to $\mathcal{A}$. The subsequence of $\mathbf{s}$ from the i-th entry to the j-th entry is denoted by $\mathbf{s}[i:j]$, where $i:j$ denote the tuple of integers $(i, i+1, \ldots, j)$. When $i > j$, by convention $i:j$ is the empty tuple. The subsequence of $\mathbf{s}$ from the i-th entry onwards is denoted by $\mathbf{s}[i:]$.

The authors are with The Chinese University of Hong Kong, Shenzhen.

A. Infinite Linear System and Zigzag Solvability

Given a sequence of symbols $\mathbf{s}$, consider a sequence $\mathbf{u}$ whose entries are the sum of some entries in $\mathbf{s}$. For $j \in \mathbb{N}_0$, suppose the jth entry $\mathbf{u}[j]$ is computed by

$$\mathbf{u}[j] = \sum_{i \in E_j} \mathbf{s}[i] \tag{1}$$

for some finite index set $E_j \in \mathbb{N}_0$. The additions in the above equation are performed in $\mathcal{A}$. The index sets E_j, for $j \in \mathbb{N}_0$, define an encoding function: $\mathbf{s} \mapsto \mathbf{u}$. We say that the infinite linear system (1) is *decodable*, or *solvable*, if the encoding function is injective on all sequences of symbols.

For practical implementation, an efficient method for solving the infinite system in (1) is preferred. One specific class of encoding functions allows a simple sequential decoding procedure as defined as follows.

Definition 1. An infinite linear system as in (1) is *zigzag solvable* if there exist two functions $f : \mathbb{N}_0 \to \mathbb{N}_0$ and $g : \mathbb{N}_0 \to \mathbb{N}_0$ such that

1) f is bijective;
2) $\{f(i)\} \subseteq E_{g(i)} \subseteq \{f(0), f(1), \dots, f(i)\}$ for all $i \in \mathbb{N}_0$.

If we can find functions f and g satisfying the requirements in Definition 1, we can recover $\mathbf{s}$ from $\mathbf{u}$. The function f specifies a decoding order for the symbols in $\mathbf{s}$. Then for $i = 0, 1, 2, \dots$, we define $\hat{\mathbf{s}}[f(i)]$ by

$$\hat{\mathbf{s}}[f(i)] := \mathbf{u}[g(i)] - \sum_{j \in E_{g(i)} \setminus \{f(i)\}} \hat{\mathbf{s}}[j].$$

We claim that the resulting sequence $\hat{\mathbf{s}}$ is the same as $\mathbf{s}$. Since f is bijective, each entry in $\mathbf{s}$ is computed once and only once. By the second condition in Definition 1, we have

$$\begin{aligned}\mathbf{u}[g(i)] &= \sum_{j \in E_{g(i)}} \mathbf{s}[j] \\ &= \mathbf{s}[f(i)] + \sum_{j \in E_{g(i)} \setminus \{f(i)\}} \mathbf{s}[j].\end{aligned}$$

for each $i \in \mathbb{N}_0$. We can now show that $\mathbf{s}[f(i)] = \hat{\mathbf{s}}[f(i)]$ for all i by an inductive argument.

We note that the function g in Definition 1 must be an injective function. Otherwise, suppose $f(i) = f(k)$ for some $i \neq k$. From $E_{g(k)} = E_{g(i)} \subseteq \{f(0), \dots, f(i)\}$, we get $k \leq i$, and from $E_{g(i)} = E_{g(k)} \subseteq \{f(0), \dots, f(k)\}$, we get $i \leq k$. This contradicts the assumption $i \neq k$.

B. Shift-Add System and Successive Solvability

In the rest of this paper we consider a special class of infinite linear systems called *shift-add systems*.

For a sequence $\mathbf{s}$ and a nonnegative integer t, the *shift operator* z^t pads t zeros in front of $\mathbf{s}$, so that $z^t\mathbf{s}$ is a function

$$\left(z^t \mathbf{s}\right)[l] = \begin{cases} 0, & \text{if } 0 \leq l < t, \\ \mathbf{s}[l-t], & \text{if } t \leq l. \end{cases}$$

When t is negative, the operator z^t represents shifting to the left. We truncate the first $|t|$ symbols in the sequence $\mathbf{s}$ and shift the remaining symbols to the left by t positions.

Definition 2 (shift operator z^t)**.** We adopt the convention that $\mathbf{s}[i] = 0$ whenever $i < 0$. For any $t \in \mathbb{Z}$, the infinite sequence $z^t\mathbf{s}$ is defined by $\left(z^t \mathbf{s}\right)[\ell] = \mathbf{s}[\ell - t]$ for $\ell \geq 0$.

A shift-add system takes n infinite sequences $\mathbf{x}_1, \dots, \mathbf{x}_n$ as inputs and produces n output sequences $\mathbf{y}_1, \dots, \mathbf{y}_n$. For $i = 1, 2, \dots, n$, the ith output sequence $\mathbf{y}_i$ is obtained by

$$\mathbf{y}_i := \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j, \tag{2}$$

where a_{ij} are integers in $\mathbb{Z}$ for $j = 1, 2, \dots, n$. The summation of sequences are performed componentwise. The integer a_{ij} represents the number of positions we shifts the sequence $\mathbf{x}_j$. We also use the convention that $z^{\infty} = 0$, the scalar zero, so that $z^{\infty}\mathbf{s}$ is the all-zero sequence. When $a_{ij} = \infty$, the sequence $\mathbf{x}_j$ is not involved in the summation in (2).

Definition 3. We call a matrix $\Phi = (z^{a_{ij}})$ a *shift matrix*, where a_{ij} are integers or ∞. A shift matrix $\Phi = (z^{a_{ij}})$ can be more conveniently represented by an integral matrix $A = (a_{ij})$. From here on, we make no distinction between Φ and the integer matrix A, and refer to both matrices as shift matrix.

Example 1. Consider the 3×3 system

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} 1 & z & z^2 \\ 1 & z^2 & z^4 \\ 1 & z^3 & z^6 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}. \tag{3}$$

We also say the shift matrix of the system is $\begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix}$.

We re-arrange the entries in the input sequences $\mathbf{x}_1, \dots, \mathbf{x}_n$ in a single sequence $\mathbf{s}$, and all entries in output sequences $\mathbf{y}_1, \dots, \mathbf{y}_n$ into another sequence $\mathbf{u}$. For $j = 1, \dots, n$ and $i \in \mathbb{N}_0$, we let

$$\mathbf{s}[in + j - 1] := \mathbf{x}_j[i], \qquad \mathbf{u}[in + j - 1] := \mathbf{y}_j[i]. \tag{4}$$

The encoding equation

$$\mathbf{y}_j[i] = \mathbf{x}_1[i - a_{j1}] + \mathbf{x}_2[i - a_{j2}] + \cdots + \mathbf{x}_n[i - a_{jn}]$$

can be represented as $\mathbf{u}[in+j-1] = \sum_{k \in E_{in+j-1}} \mathbf{s}[k]$ where

$$\begin{aligned} E_{in+j-1} = \mathbb{N}_0 \cap \{&(i - a_{j1})n, (i - a_{j2})n + 1, \dots, \\ &(i - a_{jn})n + n - 1\}. \end{aligned} \tag{5}$$

For notational convenience, we regard $\mathbf{x}_j[i]$'s and $\mathbf{y}_j[i]$'s also as the indices $in + j - 1$ in the following discussion of this section, and re-write (5) as

$$E_{\mathbf{y}_j[i]} = \left\{ \mathbf{x}_1[i - a_{j1}], \mathbf{x}_2[i - a_{j2}], \dots, \mathbf{x}_n[i - a_{jn}] \right\} \cap \mathbb{N}_0.$$

With the above notation, the definition of zigzag solvability applies to shift-add systems as well. For practical implementation, it would be more preferable if the data symbols in $\mathbf{x}_i$ can

TABLE I
SUCCESSIVE DECODING IN EXAMPLE 1. $m \geq 2$ IN THE LAST THREE ROWS

Iteration i	$f(i)$	Equation used in solving $f(i)$
0	$\mathbf{x}_1[0]$	$\mathbf{y}_3[0] = \mathbf{x}_1[0]$
1	$\mathbf{x}_1[1]$	$\mathbf{y}_3[1] = \mathbf{x}_1[1]$
2	$\mathbf{x}_1[2]$	$\mathbf{y}_3[2] = \mathbf{x}_1[2]$
3	$\mathbf{x}_2[0]$	$\mathbf{y}_2[2] = \mathbf{x}_2[0] + \mathbf{x}_1[2]$
$3m-2$	$\mathbf{x}_1[m+1]$	$\mathbf{y}_3[m+1] = \mathbf{x}_1[m+1] + \mathbf{x}_2[m-2] + \mathbf{x}_3[m-5]$
$3m-1$	$\mathbf{x}_2[m-1]$	$\mathbf{y}_2[m+1] = \mathbf{x}_1[m+1] + \mathbf{x}_2[m-1] + \mathbf{x}_3[m-3]$
$3m$	$\mathbf{x}_3[m-2]$	$\mathbf{y}_1[m] = \mathbf{x}_1[m] + \mathbf{x}_2[m-1] + \mathbf{x}_3[m-2]$

TABLE II
ZIGZAG DECODING IN EXAMPLE 2. $m \geq 2$ IN THE LAST THREE ROWS

Iteration i	$f(i)$	Equation used in solving $f(i)$
0	$\mathbf{x}_2[1]$	$\mathbf{y}_2[0] = \mathbf{x}_2[1]$
1	$\mathbf{x}_1[1]$	$\mathbf{y}_1[1] = \mathbf{x}_1[1] + \mathbf{x}_2[1]$
2	$\mathbf{x}_2[0]$	$\mathbf{y}_3[0] = \mathbf{x}_2[0] + \mathbf{x}_1[1]$
3	$\mathbf{x}_1[0]$	$\mathbf{y}_1[0] = \mathbf{x}_1[0] + \mathbf{x}_2[0]$
$3m-2$	$\mathbf{x}_2[m]$	$\mathbf{y}_2[m-1] = \mathbf{x}_2[m]$
$3m-1$	$\mathbf{x}_1[m]$	$\mathbf{y}_1[m] = \mathbf{x}_1[m] + \mathbf{x}_2[m]$
$3m$	$\mathbf{x}_3[m-2]$	$\mathbf{y}_3[m-1] = \mathbf{x}_1[m] + \mathbf{x}_2[m-1] + \mathbf{x}_2[m-2]$

be decoded in the order $\mathbf{x}_i[0]$, $\mathbf{x}_i[1]$, $\mathbf{x}_i[2], \dots$ We formulate a stronger notion of solvability as follows.

Definition 4 (successive solvability)**.** A shift-add system in (2) is *successively solvable* if it is zigzag solvable and the functions f and g in Definition 1 satisfy an additional condition:

3) $f^{-1}(\mathbf{x}_j[\ell_1]) < f^{-1}(\mathbf{x}_j[\ell_2])$ whenever $0 \leq \ell_1 < \ell_2$, for all $j = 1, 2, \dots, n$.

The last condition in Definition 4 guarantees that symbol $\mathbf{x}_j[\ell_1]$ is computed before $\mathbf{x}_j[\ell_2]$ if $\ell_1 < \ell_2$. For a successively solvable shift-add system, there exists a zigzag solving algorithm that solves the symbols of each variable sequence from "left" to "right". That is, the algorithm will solve the symbols of $\mathbf{x}_i$ in the order $\mathbf{x}_i[0]$, $\mathbf{x}_i[1]$, $\mathbf{x}_i[2], \dots$ This definition of successive solvability is general enough to include the existing algorithms for solving special shift-add systems in literature [1], [2], [4], [5], [10].

For example, the shift-add system in Example 1 is successively solvable. The order of decoding the data symbols are listed in Table I. However, a zigzag solvable system is not necessarily successively solvable. We present a counter-example below.

Example 2. Consider the shift-add system given by the shift matrix

$$\begin{pmatrix} 0 & 0 & \infty \\ \infty & -1 & \infty \\ -1 & 0 & 1 \end{pmatrix}.$$

This shift-add system is zigzag solvable. The order of decoding the symbols are shown in Table II. Note that the definition of successive decoding is violated. In fact, there is no way to successively decode the system. We have to first decode $\mathbf{x}_2[\ell]$, for some $\ell \geq 1$ to kick-start the zigzag decoding procedure.

The shift-add system in Example 2 is somewhat pathological. However, the next proposition states that for systems satisfying some mild conditions, the two notions of solvability are equivalent.

Proposition 1. *For a shift-add system corresponding to a non-negative shift matrix, it is zigzag solvable if and only if it is successively solvable.*

Proof. ($\Leftarrow$). This part is trivial since the definition of successive solvability requires zigzag solvability.

($\Rightarrow$). Suppose on the contrary that one such system is zigzag solvable but not successive solvable. Let a zigzag decoding algorithm of this system be given by functions f and g satisfying the conditions in Definition 1. We order the input and output symbols as in (4). The sets E_k that specify the encoding function are denoted by $E_{\mathbf{y}_j[i]}$, by identifying $\mathbf{y}_j[i]$ with the index $k = ni + j - 1$.

Let a be the smallest number where there exists $1 \leq \nu \leq n$ and $\ell' < \ell$ such that

$$a := f^{-1}(\mathbf{x}_\nu[\ell]) < f^{-1}(\mathbf{x}_\nu[\ell']) =: a'. \tag{6}$$

Such an a exists because it is assumed that the system is not successively solvable. Let i and j be integers such that $g(a) = E_{\mathbf{y}_j[i]}$, then we have

$$E_{\mathbf{y}_j[i+\ell'-\ell]} = \{\mathbf{x}_u[v+\ell'-\ell] : \mathbf{x}_u[v] \in E_{\mathbf{y}_j[i]},\ v+\ell'-\ell \geq 0\}.$$

By the minimality of a, we have that $f^{-1}(\mathbf{x}_u[v + \ell' - \ell]) < f^{-1}(\mathbf{x}_u[v]) < a$ for all $\mathbf{x}_u[v] \in E_{\mathbf{y}_j[i]}$ and $\mathbf{x}_u[v] \neq \mathbf{x}_\nu[\ell]$. Thus, at iteration a, we can solve $\mathbf{x}_i[\ell']$ instead of $\mathbf{x}_i[\ell]$, and exchange the order of $\mathbf{x}_i[\ell]$ and $\mathbf{x}_i[\ell']$ in the decoding order defined by the function f.

After the interchange, if the resulting decoding order satisfies the requirement of successive decoding, then it violates the assumption that the system is not successively solvable. We then find the next smallest number a such that (6) is satisfied for some other ℓ and ℓ'. This value of a is strictly larger than the value of a in the previous paragraph. We can repeat the argument in the previous paragraph and produce a decoding order that satisfies the criterion of successive solvability. $\square$

III. REDUCTION OF SHIFT-ADD SYSTEMS

In this section we introduce some notations that can help to described the transition of the shift matrices when solving the corresponding shift-add systems successively. Parts of the discussion here can also be found in literature, e.g., [19], but the systematic treatment of equivalence and reducing operations is new. To motivate the definitions to be introduced in this section, we first consider a 3×3 example.

Example 3. Consider the following shift-add system.

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} z^{-2} & z^{-1} & 1 \\ z^{-2} & 1 & z^2 \\ 1 & z^3 & z^6 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}, \tag{7}$$

where $\mathbf{x}_i$, for $i = 1, 2, 3$, are input sequences, and $\mathbf{y}_i$, for $i = 1, 2, 3$, are the output sequences. The first two symbols in the sequence $\mathbf{x}_1$ are truncated in $\mathbf{y}_1, \mathbf{y}_2$ and appear in the linear system only through $\mathbf{y}_3$.

This system is successively solvable. We first solve the first 3 symbols in $\mathbf{x}_1$ using the first 3 symbols in $\mathbf{y}_3$. After subtracting $\mathbf{x}[0]$, $\mathbf{x}[1]$ and $\mathbf{x}[2]$ from $\mathbf{y}_1$ and $\mathbf{y}_2$, the linear system is reduced to

$$\begin{bmatrix} \mathbf{y}_1' \\ \mathbf{y}_2' \\ \mathbf{y}_3' \end{bmatrix} = \begin{bmatrix} z & z^{-1} & 1 \\ z & 1 & z^2 \\ 1 & 1 & z^3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1[3:] \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}. \tag{8}$$

The sequence $\mathbf{y}_1'$ and $\mathbf{y}_2'$ and $\mathbf{y}_3'$ are obtained by

$$\mathbf{y}_1'[i] := \begin{cases} \mathbf{y}_1[i] - \mathbf{x}_1[0] & \text{if } i = 0, \\ \mathbf{y}_1[i] & \text{if } i \geq 1, \end{cases}$$

$$\mathbf{y}_2'[i] := \begin{cases} \mathbf{y}_2[i] - \mathbf{x}_1[0] & \text{if } i = 0, \\ \mathbf{y}_2[i] & \text{if } i \geq 1, \end{cases}$$

$$\mathbf{y}_3'[i] := \mathbf{y}_3[i+3], \quad \text{for } i \in \mathbb{N}_0.$$

Note that the shift matrix in (8) is obtained from the shift matrix in (7) by multiplying the first column by z^3 and then dividing the third row by z^{-3}. Correspondingly, $\mathbf{x}[0]$, $\mathbf{x}[1]$ and $\mathbf{x}[2]$ as well as $\mathbf{y}[0]$, $\mathbf{y}[1]$ and $\mathbf{y}[2]$ are removed from the linear system.

We can then obtain the first symbol in $\mathbf{x}_2$ from the first symbol of $\mathbf{y}_2'$. We subtract the value of $\mathbf{x}_2[0]$ from $\mathbf{y}_1'$ and $\mathbf{y}_3'$, and transform the linear system to

$$\begin{bmatrix} \mathbf{y}_1'' \\ \mathbf{y}_2'' \\ \mathbf{y}_3'' \end{bmatrix} = \begin{bmatrix} z & 1 & 1 \\ 1 & 1 & z \\ 1 & z & z^3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1[3:] \\ \mathbf{x}_2[1:] \\ \mathbf{x}_3 \end{bmatrix}. \tag{9}$$

The sequence $\mathbf{y}_1''$ and $\mathbf{y}_2''$ and $\mathbf{y}_3''$ are obtained by

$$\mathbf{y}_1''[i] := \mathbf{y}_1'[i] \quad \text{for } i \in \mathbb{N}_0,$$

$$\mathbf{y}_2''[i] := \mathbf{y}_2'[i+1] \quad \text{for } i \in \mathbb{N}_0,$$

$$\mathbf{y}_3''[i] := \begin{cases} \mathbf{y}_3'[0] - \mathbf{x}_[0] & \text{if } i = 0 \\ \mathbf{y}_3'[1] & \text{if } i \geq 1. \end{cases}$$

The shift matrix in (9) is obtained from the shift matrix in (8) by multiplying the second column by z and then dividing the second row by z.

From the first symbols in $\mathbf{y}_3''$, $\mathbf{y}_1''$ and $\mathbf{y}_2''$ we can obtain the first symbols in $\mathbf{x}_1[3:]$, $\mathbf{x}_2[1:]$, and $\mathbf{x}_3$, by solving a system of three equations that can be permuted to an lower triangular form,

$$\begin{bmatrix} \mathbf{y}_3''[0] \\ \mathbf{y}_2''[0] \\ \mathbf{y}_1''[0] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1[3] \\ \mathbf{x}_2[1] \\ \mathbf{x}_3[0] \end{bmatrix}.$$

After subtracting $\mathbf{x}_1[3]$, $\mathbf{x}_2[1]$ and $\mathbf{x}_3[0]$, the linear system reduces to another linear system with the same shift matrix as in (9). We can henceforth successively decode the rest of the data symbols.

In view of the previous example we make the following definitions.

Definition 5. We denote the ith row of shift matrix A by $\mathbf{a}_i$, and an all-one row vector by $\mathbf{1}$. For two row vectors $\mathbf{a} = (a_j)$ and $\mathbf{b} = (b_j)$ of the same length with integer components, we write $\mathbf{a} \succeq \mathbf{b}$ if $a_j \geq b_j$ for all j. In particular, for a constant c, we write $\mathbf{a} \succeq c$ if $\mathbf{a} \succeq c\mathbf{1}$.

Definition 6. We say that two $n \times n$ shift matrices A and B are *equivalent* if for each $i = 1, 2, \ldots, n$, we have

$$\mathbf{a}_i = \mathbf{b}_i + c_i \mathbf{1}$$

for some integers $c_1, c_2, \ldots, c_n$, with $c_i = 0$ whenever $\mathbf{a}_i$ or $\mathbf{b}_i$ contain some negative entries. We adopt the convention $\infty + c = \infty - c = \infty$ for any integer c. We write $A \sim B$ if A is equivalent to B.

The relation defined in Definition 6 is an equivalence relation on all $n \times n$ shift matrices. Any two equivalent shift matrices define the same infinite linear system.

Example 4. The shift matrices

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 2 \\ 3 & 3 & 6 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

are equivalent. They represent the same system as in (8).

Definition 7. The equivalence class of matrix A is denoted as $\{A\}$. A shift matrix A with $\min(\mathbf{a}_i) = 0$ whenever $\mathbf{a}_i \succeq 0$ is said to be in *canonical form*. Each equivalence class includes a unique matrix A in canonical form, which is called the (canonical) *representative* of the equivalence class.

We can assume that a shift matrix does not contain any row whose entries are all ∞, otherwise the corresponding output sequence is the all-zero sequence and contains no information about the input symbol sequences. Hence the canonical representative of the equivalence class is well-defined. From now on, when we talk about a shift matrix, we usually refer to the canonical representative of its equivalence class.

Example 5. A 1×1 shift matrix A is successively solvable if and only if the entry is non-negative. Therefore, successively solvable 1×1 shift matrices are precisely the shift matrices contained in the equivalence class $\{[0]\}$.

Definition 8. Given a shift matrix A, the (i, j)-entry is called a *pivot* if (i) $a_{ij} \geq 0$, and (ii) $a_{ij} < a_{ij'}$ for all $j' \neq j$. We say that a shift matrix A is *reductive* if A contains a pivot.

In other words, we say that the (i, j)-entry is a pivot of A if and only if a_{ij} is the *strict minimum* in $\mathbf{a}_i$ and $a_{ij} \geq 0$.

For a shift matrix A with the (i, j)-entry as a pivot, the first symbol in the ith output sequence is precisely equal to the first symbol in the jth input sequence. We can thus solve the first symbol in the jth input sequence by reading the first symbol in the ith output sequence, subtract it from the linear system, and reduce the system to another linear system. We call this process *reducing operation with respect to the ith row*. For instance, in Example 4, the $(2, 2)$-entry is a pivot, and we can directly read off the first symbol in sequence $\mathbf{x}_2$ from the first symbol in $\mathbf{y}_2$. Meanwhile there is no pivot in the first and third row.

After a reducing operation, the shift matrix is modified to another shift matrix. The difference between the two shift matrices before and after a reducing operation is given by the matrix in the following:

Definition 9 (reducing operator)**.** For $k, \ell \in \{1, 2, \dots, n\}$, we define the $n \times n$ matrix $R_{k\ell} = (\delta_{j,\ell} - \delta_{i,k})$ where $\delta_{i,k}$ denotes the Kronecker's delta function.

The matrix $R_{k\ell}$ can be obtained from the zero matrix by adding 1 to each of the entries in the ℓth column and subtracting 1 from each of the entries in the kth row. If a shift matrix A is reductive with the (k, ℓ)-entry as a pivot, we obtain the matrix $R_{k\ell} + A$ if we perform a reducing operation with respect to the kth row.

Lemma 2. *Suppose the (k, ℓ)-entry of an $n \times n$ shift matrix A is a pivot. If $A' \sim A$, then the (k, ℓ)-entry is a pivot of A', and $(R_{k\ell} + A) \sim (R_{k\ell} + A')$.*

Proof. The difference between the kth row $\mathbf{a}_k$ of A' and the kth row $\mathbf{a}'_k$ of A is equal to $c\mathbf{1}$ for some constant c. Hence the ℓth component in $\mathbf{a}'_k$ is a strict minimum if and only if the ℓth component in $\mathbf{a}_k$ is a strict minimum. This shows that the (i, j)-entry in A' is a pivot of A'. For the second part of (b), we note that $\mathbf{a}_k \succeq 0$ if the there is a pivot in the kth row of A, and

$$(R_{k\ell} + A) - (R_{k\ell} + A') = A - A'.$$

The ith row of $(R_{k\ell} + A) - (R_{k\ell} + A')$ is in the form $c_i\mathbf{1}$ for some constant c_i. The constant c_i is zero whenever the ith row of $(R_{k\ell} + A)$ or $(R_{k\ell} + A')$ contain some negative entry, because in this case the ith row of A or A' also contain some negative entry. □

Lemma 2 implies that we can define the reducing operation on the equivalence classes of shift matrices.

Example 6. In Example 1, the processing of solving the shift-add system in (3) can be expressed in terms of the matrices $R_{k\ell}$'s as follows. The initial shift matrix is

$$A_0 = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix}.$$

We apply the reducing operation with respect to the 3rd row three times. The shift matrix becomes

$$A_1 := R_{31} + R_{31} + R_{31} + A_0 = \begin{bmatrix} 3 & 1 & 2 \\ 3 & 2 & 4 \\ 0 & 0 & 3 \end{bmatrix},$$

which is exactly the shift matrix of the system after three iterations in Table I. The $(2, 2)$-entry in A_1 is pivot, and we perform a reducing operation on the second row. We obtain

$$A_2 := R_{22} + A_1 = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 2 & 3 \\ 0 & 1 & 3 \end{bmatrix},$$

which is the shift matrix of the system after iteration 3 in Table I. The decoding process continues by repeatedly applying reducing operations on the 3rd row, the 2nd row and the 1st row. The shift matrices after these three reducing operations does not change: $A_2 = R_{13} + R_{22} + R_{31} + A_2$. The successively solving algorithm can be represented by a sequence of reducing operators:

$$R_{31}, R_{31}, R_{31}, R_{22}, \underline{R_{31}, R_{22}, R_{13}}, \dots$$

repeating the underlined part *ad infinitum*.

In general the order of the reducing operators cannot be changed. For instance, in Example 6, the order of applying the reducing operations to row 3, row 2 and row 1 is important. It is not valid if we apply the reducing operations in the order of row 1, row 2 and row 3, because we are not able to find the appropriate pivot in each step. However, if there are two pivots in a shift matrix already, the order of applying the corresponding reducing operations is irrelevant.

Lemma 3. *Suppose an $n \times n$ shift matrix A contains two pivots at locations (k_1, ℓ_1) and (k_2, ℓ_2) with $\ell_1 \neq \ell_2$. Then we can apply the reducing operations with respect to the k_1th row or the k_2th row in any order, and the shift matrix after these two operations are the same.*

Proof. We necessarily have $k_1 \neq k_2$ because a row cannot contain two pivots. If we perform a reducing operation at row k_1, the pivot at (k_2, ℓ_2) continue to be a pivot after the operation. On the other hand, if we perform a reducing operation with respect to row k_2 first, the pivot at (k_1, ℓ_1) is not destroyed after the operation. The resulting shift matrices after these two operations

$$R_{k_1\ell_1} + R_{k_2\ell_2} + A \text{ and } R_{k_2\ell_2} + R_{k_1\ell_1} + A$$

are actually equal to each other. □

IV. Graphical Characterization of Successively Solvability

In this section we consider shift-add systems whose shift matrices do not contain the ∞ symbol, i.e., each output sequence is a function of all input sequences.

Definition 10. For a fixed positive integer n, we define a directed graph $\mathcal{G}_n$ whose vertices are equivalence classes of shift matrices with finite entries. Given two shift matrices A and B with finite entries, there is a directed edge from $\{A\}$ to $\{B\}$ if A has a pivot (k, ℓ) and $R_{k\ell} + A \sim B$.

For the ease of notation, we write $A \xrightarrow{R_{k\ell}} B$ if there is a directed edge from $\{A\}$ to $\{B\}$ with respect to the (k, ℓ) pivot in A. It follows from definition that there is no self-loop in $\mathcal{G}_n$. When there are multiple pivots in the same column of A, there are multiple edges from $\{A\}$ to $\{B\}$

Definition 11. For positive integer L, we define a *path* of length L in $\mathcal{G}_n$ as a sequence of edges $A_k \xrightarrow{R_{i_k j_k}} B_k$, $k = 1, 2, \dots, L$, where $B_k \sim A_{k+1}$ for $k = 1, 2, \dots, L-1$. When $B_L \sim A_1$, the path is called a *cycle*. Given two shift matrices, A and B, we say that a vertex $\{B\}$ is *reachable* from vertex $\{A\}$ in $\mathcal{G}_n$ if there is a directed path from $\{A\}$ to $\{B\}$.

$$
\begin{tikzcd}
\begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix} \arrow[d, "R_{31}"'] & \begin{bmatrix} 3 & 1 & 2 \\ 3 & 2 & 4 \\ 0 & 0 & 3 \end{bmatrix} \arrow[r, "R_{22}"] & \begin{bmatrix} 3 & 2 & 2 \\ 2 & 2 & 3 \\ 0 & 1 & 3 \end{bmatrix} \arrow[d, "R_{31}"'] & \begin{bmatrix} 4 & 3 & 2 \\ 2 & 2 & 2 \\ 0 & 1 & 2 \end{bmatrix} \arrow[l, "R_{13}"'] \arrow[d, "R_{31}"'] \\
\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 4 \\ 0 & 2 & 5 \end{bmatrix} \arrow[r, "R_{31}"] & \begin{bmatrix} 2 & 1 & 2 \\ 2 & 2 & 4 \\ 0 & 1 & 4 \end{bmatrix} \arrow[u, "R_{31}"'] & \begin{bmatrix} 4 & 2 & 2 \\ 3 & 2 & 3 \\ 0 & 0 & 2 \end{bmatrix} \arrow[ur, "R_{22}"] & \begin{bmatrix} 5 & 3 & 2 \\ 3 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix} \arrow[l, "R_{13}"']
\end{tikzcd}
$$

Fig. 1. Successively solving the shift-add system in Example 1. Pivots are circled.

Remark. A path in this paper need not be simple, i.e., a path may contain repeated vertices. When we refer to a path consisting of distinct vertices, we will emphasize it by saying *simple path*.

The successively solving algorithm of the system (3) discussed in Example 6 can be illustrated as a subgraph of $\mathcal{G}_3$ in Fig. 1. Although the graph $\mathcal{G}_n$ is an infinite graph, the degree is finite, and the arguments in what follows can be done on a finite part of it.

Lemma 4. *For each shift matrix A, the set of vertices in $\mathcal{G}_n$ that are reachable from $\{A\}$ is finite.*

Proof. Let A' be the canonical representation of the equivalence class $\{A\}$. If A' is not reductive, then the lemma trivially holds.

Suppose $A \to B_1 \to B_2 \to \cdots \to B_L$ is a path in $\mathcal{G}_n$. For $t = 1, 2, \ldots, L$, let B'_t be the canonical representation of B_t. In each step from B_t to B_{t+1}, the shift matrix B'_{t+1} is obtained from B'_t by first adding a matrix in the form R_{ij} for some i and j in $\{1, 2, \ldots, n\}$. If the resulting matrix contains a row that consists solely of positive entries, we can subtract 1 from each entry in that row, until we arrive at a shift matrix in canonical form. During this process, the total sum of all entries in the matrix can only decrease.

Let the magnitude of the smallest negative entries be denoted by μ. Otherwise, we can set μ to be zero. Therefore, each of the shift matrices $B'_1, \ldots, B'_L$ belongs to the set of matrices over $\mathbb{Z}$ whose entries are all larger than $-\mu$ and the sum of all entries is less than the sum of entries in A'. This set is certainly finite. hence $\{A\}$ can reach finitely many distinct vertices in $\mathcal{G}_n$. $\square$

Lemma 5. *Suppose $A_1 \to A_2 \to \cdots \to A_L \to A_{L+1} \sim A_1$ is a cycle of length L in $\mathcal{G}_n$, and the (i_t, j_t)-entry in A_t is used as the pivot in the transition from A_t to A_{t+1}, for $t = 1, 2, \ldots, L$. Then*

(a) *the length of cycle L is a multiple of n;*
(b) *all entries in $A_1, A_2, \ldots, A_L$ are nonnegative;*
(c) *If A_1 is the canonical representation of the equivalence class $\{A_1\}$,*

$$R_{i_t j_t} + R_{i_{t-1} j_{t-1}} + \cdots + R_{i_2 j_2} + R_{i_1 j_1} + A_1 \tag{10}$$

is the canonical representation of $\{A_{t+1}\}$, for $t = 1, 2, \ldots, L$.

Proof. By the assumption that $A_1 \to A_2 \to \cdots \to A_L \to A_1$ is a cycle, the matrix

$$\sum_{t=1}^{L} R_{i_t, j_t} + A_1 \tag{11}$$

is similar to A_1. Hence we can write

$$\sum_{t=1}^{L} R_{i_t j_t} = \text{diag}(c_1, c_2, \ldots, c_n) J,$$

where J is an $n \times n$ all-one matrix, for some constants $c_1, \ldots, c_n$. This means that the entries in each row of $\sum_{t=1}^{L} R_{i_t j_t}$ are constant. Recall that the effect of adding $R_{i_t j_t}$ is to add 1 to all entries in column j_t and subtract 1 from all entries in row i_t. The equality in (11) can hold only when we add equal numbers of 1's to each of the n columns. This proves part (a).

If A_1 contains a negative entry, say at row k, then, by the definition of equivalence of shift matrices, the value of c_k must be zero. In other words, the kth row in (11) is identical to the kth row in A_1. Because the only way to decrease the values of the entries in the kth row is by applying a reducing a operation to the kth row, we must have a pivot in the kth row in one of $A_2, \ldots, A_L$. After a reducing operation with respect to the kth row, the entries in the kth row will be nonnegative. In particular, the entries in the kth row of (11) are all nonnegative. This contradicts the assumption that A_1 contains a negative entry in the kth row. By picking a different start point in the cycle, we can similar show that there is no nonegative entry in all shift matrices A_1 to A_L.

For part (c), suppose A_1 is the canonical representative of $\{A_1\}$. This means that all entries in A_1 are nonnegative and the minimum value in each row of A_1 is equal to zero. We now use the fact that a reducing operation does not decrease the minimum value in each row of A, i.e., the minimum value can only increase or remain unchanged. For $t = 1, 2, \ldots, L$, the minimum value in any row in the shift matrix in (10) is larger than or equal to zero. When $t = L$, the shift matrix in (10) is the same as A_1, which is canonical. Therefore the minimum values in the rows of (10) must be all zero. $\square$

An immediate consequence of the previous lemma is that, for $i = 1, 2, \ldots, n$, exactly L/n pivots in the the list $((i_t, j_t))_{t=1}^{L}$ are contained in the ith row, and exactly L/n of them in the ith column. Indeed, we can strengthen this property in the following theorem.

Theorem 6. *For a cycle in $\mathcal{G}_n$, there is a permutation σ of $\{1, 2, \ldots, n\}$ such that the reducing operations in the cycle are represented by $R_{i\sigma(i)}$ for $i = 1, \ldots, n$.*

Proof. By part (c) of Lemma 5, we can consider without loss of generality a cycle

$$A_1 \to A_2 \to \cdots \to A_L \to A_1$$

with A_t being the canonical representation of $\{A_t\}$ for all $t = 1, 2, \ldots, L$. We can express A_{t+1} by the summation in (10).

Suppose the pivot of the reducing operation transforming A_t to A_{t+1} is located at (i_t, j_t), for $t = 1, 2, \ldots, L$. Since A_t is canonical, the (i_t, j_t)-entry in A_t is equal to zero. We want to show that there is a permutation σ of $\{1, 2, \ldots, n\}$ such that $j_t = \sigma(i_t)$ for all t.

We prove the theorem by contradiction. Suppose there exists two distinct indices α and $\bar{\alpha}$, with $1 \leq \alpha < \bar{\alpha} \leq L$, such that $i_\alpha \neq i_{\bar{\alpha}}$, $j_\alpha = j_{\bar{\alpha}} = \ell$. Because the shift matrix A_{i_α} are canonical, the ℓth entry in the (i_α)th row of A_α is zero. Immediately after the transition from A_α to $A_{\alpha+1}$, the ℓth entry in all rows, except row i_α, are larger than or equal to 1. If the shift matrix $A_{\bar{\alpha}}$ has the $(i_{\bar{\alpha}}, \ell)$-entry as a pivot at iteration $\bar{\alpha}$, we must apply a reducing operation with respect to row $i_{\bar{\alpha}}$ before iteration $\bar{\alpha}$. Suppose β is a time index strictly between α and $\bar{\alpha}$ such that $i_\beta = i_{\bar{\alpha}}$ and $j_\beta = \ell' \neq \ell$.

However, at iteration $\bar{\alpha}$, the (i_β, ℓ')-entry is no longer a pivot in $A_{i_{\bar{\alpha}}}$. We must have have another time indices $\bar{\beta}$ strictly between β and $\bar{\alpha}$ such that $j_{\bar{\beta}} = \ell'$. We thus find another pair of indices β and $\bar{\beta}$ that satisfies $\alpha < \beta < \bar{\beta} < \bar{\alpha}$ and $j_\beta = j_{\bar{\beta}} = \ell'$. Indeed, we can repeat the whole argument arbitrarily many times, but this is not possible as all the time indices are integers bounded between 1 and L.

This shows that the mapping σ defined by $i_t \mapsto j_t$ is one-to-one from $\{1, 2, \ldots, L\}$ to itself. Therefore it must be a bijection. □

Theorem 7. *Let $\{A\}$ be a equivalence class of shift matrices contained in a cycle of $\mathcal{G}_n$. If there is a directed path from $\{A\}$ to $\{B\}$ for some shift matrix B, then there is a directed path from $\{B\}$ to $\{A\}$, i.e., $\{A\}$ and $\{B\}$ are contained in a cycle.*

Proof. It suffices to prove the case when there is a directed edge from A to B. Suppose $A = A_1$ and

$$A = A_1 \to A_2 \to \cdots \to A_L \to A$$

is a cycle in $\mathcal{G}_n$, and for $t = 1, 2, \ldots, L$, the (i_t, j_t)-entry in A_t is used as a pivot in reducing A_t. If $B \sim A_2$, then the theorem holds trivially.

Assume that $\{B\}$ and $\{A_2\}$ are two distinct equivalence classes of shift matrices. Since the shift matrix A can be reduced to shift matrices B and A_2, there must be at least two pivots in A. We already know that the (i_1, i_2)-entry is a pivot. Suppose $B = R_{k\ell} + A_1$, and the (k, ℓ)-entry is another pivot in A_1. Since each row contains at most one pivot, we have $k \neq i_1$. We also have $\ell \neq j_1$. Otherwise, if $\ell = j_1$, then $B \sim A_2$, contrary to the assumption that $\{B\}$ and $\{A_2\}$ are two distinct equivalence classes.

By part (c) of Lemma 5, there is an index $t \in \{1, \ldots, L\}$ such that $j_t = \ell$. We take the smallest such t if there are more than one t with $j_t = \ell$. The shift matrix

$$A_t = R_{i_{t-1}j_{t-1}} + \cdots + R_{i_2 j_2} + R_{i_1 j_1} + A$$

is reductive with the (i_t, j_t)-entry as a pivot. By Theorem 6, we must have $(i_t, j_t) = (k, \ell)$. We now create a new cycle

$$A \xrightarrow{R_{i_t,j_t}} B_2 \xrightarrow{R_{i_1,j_1}} B_3 \xrightarrow{R_{i_2,j_2}} B_4 \to \cdots \xrightarrow{R_{i_{t-1},j_{t-1}}} B_{t+1}$$
$$\xrightarrow{R_{i_{t+1},j_{t+1}}} B_{t+2} \to \cdots \xrightarrow{R_{i_L,j_L}} A.$$

by permuting the order of the reducing operations. Since $(i_t, j_t) = (k, \ell)$, we have $B_2 = B$. By out choice of i_t, we have $j_a \neq \ell$ for $a = 1, 2, \ldots, t-1$. We can apply Lemma 3 $(t-1)$ times to show that the shift matrices B_2 to B_t are reductive and the reducing operations written in the above cycle are valid. After the reducing operation $R_{i_{t-1},j_{t-1}}$ in the new cycle, the shift matrices $B_{t+1}, B_{t+2}, \ldots$ are the same as $A_{t+1}, A_{t+2}, \ldots$, respectively. We thus produce another cycle from A to B and then back to A. □

Definition 12. A directed graph $\mathcal{H}$ in general is said to be *strongly connected* if for any two vertices v and u in $\mathcal{H}$, there is a directed path from v to u and a directed path from u to v. A *strongly connected component* is a maximal strongly connected subgraph. In the graph $\mathcal{G}_n$, a strongly connected component can be written as a union of cycles of vertices and is called a *cycle component*. The permutation σ defined in Theorem 6 is uniquely defined throughout a cycle component, and is called the *character* of the cycle component.

Theorem 7 says that for two equivalence classes of shift matrices $\{A\}$ and $\{A'\}$ in different cycle components, there exists no directed paths between $\{A\}$ and $\{A'\}$ in $\mathcal{G}_n$. This property is extended in the next theorem.

Theorem 8. *Suppose $\{B\}$ and $\{C\}$ are two vertices contained in two distinct cycle components in $\mathcal{G}_n$, and there is a directed path from $\{A\}$ to $\{B\}$. Then there is no directed path from $\{A\}$ to the cycle component containing $\{C\}$.*

Proof. We first prove a base case. Suppose B can be reached from A in one step and C can also be reached from A in one step, i.e.,

$$A \xrightarrow{R_{ij}} B, \quad \text{and} \quad A \xrightarrow{R_{k\ell}} C$$

for some i, j, k and ℓ. Since B and C are not similar, the shift matrix A thus contains two distinct pivots with $i \neq k$ and $j \neq \ell$. By Lemma 3, we can apply reducing operations to A with repect to row i and row j in any order. The shift matrix $R_{ij} + R_{k\ell} + A$ is reachable from both B and C. This violates Theorem 7 because it is assumed that B and C are contained in two difference cycle components.

We apply double mathematical induction in the general case. Given a shift matrix A, let $d_1(A)$ (resp. $d_2(A)$) be the length of a shortest simple directed from A to the cycle component that contains $\{B\}$ (resp. $\{C\}$). By Lemma 4, any simple path starting with vertex $\{A\}$ must be finite. We let $d_2(A) = \infty$ if the vertices in the cycle component that contains C are not reachable from A. By assumption, $d_1(A)$ is finite. We want to show that $d_2(A) = \infty$.

We have shown in the previous paragraph that the case $d_1(A) = d_2(A) = 1$ is impossible. Suppose $d_1(A) = 1$ and $d_2(A) = 2$, i.e.,

$$A \xrightarrow{R_{ij}} B, \quad \text{and} \quad A \xrightarrow{R_{k_1\ell_1}} A_1 \xrightarrow{R_{k_2\ell_2}} C.$$

By the same argument as in the first paragraph in the proof, we have $i \neq k_1$ and $j \neq \ell_1$. We can thus compute the order in applying the reducing operations associated with R_{ij} and $R_{k_1\ell_1}$. Then the matrix

$$R_{k_1\ell_1} + R_{ij} + A$$

is a shift matrix in the same cycle component as B, and is reachable by A_1 in one step. Hence $d_1(A_1) = d_2(A_1) = 1$, which cannot be true by the previous paragraph.

The proof is finished by a double mathematical induction. $\square$

Definition 13. Given a cycle component in $\mathcal{G}_n$, we define its *extended cycle component* be the set of vertices $\{A\}$ in $\mathcal{G}_n$ that connects to the given cycle component, i.e., there is a directed path from $\{A\}$ to any vertex in the cycle component.

We note that the notion of extended cycle component is well-defined by Theorem 8. Summarizing all the results in this section, we have the following theorem:

Theorem 9. *An $n \times n$ shift-add system defined by shift matrix A with only finite entries is successively solvable if and only if $\{A\}$ is contained in an extended cycle component of $\mathcal{G}_n$.*

Proof. Suppose A has size n. If $\{A\}$ is in a cycle component, then by Lemma 5 there exists a sequence of kn, where k is an integer, reducing operations that transforms A to one in $\{A\}$. As these reducing operations satisfies the property in Theorem 6, these reducing operations can solve k symbols from the beginning of each sequence in the corresponding shift-add system. If we repeat this sequence of reducing operations, a successive solving algorithm can be obtained satisfying Definition 4. If $\{A\}$ is not in a cycle component, there exists a directed path from $\{A\}$ to a vertex in a cycle component. The path is of a finite length by Lemma 4, and hence induces a finite number of reducing operations that transform A to A' such that $\{A'\}$ is in a cycle component. As A' is successively solvable, so does A.

To prove the necessary condition, if A is successively solvable, then there exists a sequence of infinite reducing operations that can be applied on A successively, which forms an arbitrarily long path in $\mathcal{G}_n$. As the number of vertices in $\mathcal{G}_n$ can be reached by $\{A\}$ is finite (Lemma 4), $\{A\}$ must reach certain vertex $\{A'\}$ twice in the path. Therefore, $\{A'\}$ is in a cycle component and hence $\{A\}$ is in an extended cycle component. $\square$

We illustrate an extended cycle component in $\mathcal{G}_2$ in Fig. 2. The integer a and b are positive integers. The dashed arrow indicate a directed edge belong to an extended cycle component by not in a cycle component, and the superscript * means that the corresponding reducing operations are applied multiple times. We note that the shift matrices in the middle has two pivots, and hence we can go to the left or the right from this shift matrix. The shift-add system associated to the shift matrices in Fig. 2 are all successively solvable. The character of the cycle component is $\sigma(1) = 1$ and $\sigma(2) = 2$. We only see R_{11} and R_{22} within the cycle component, but we may have other types of reducing operations outside the cycle component.

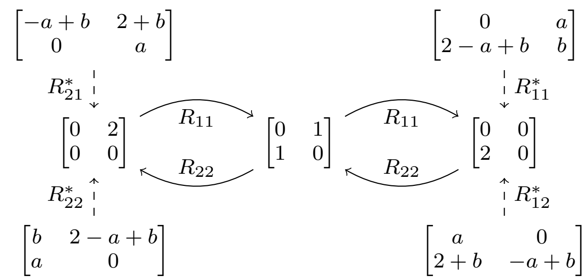

Fig. 2. An extended cycle component in $\mathcal{G}_2$. The dashed arrow indicate a directed edge belong to an extended cycle component by not in a cycle component, and the superscript * means that the corresponding reducing operations are applied multiple times.

V. Concluding Remarks

Our characterization of successively solvable shift matrices also sheds some light on shift matrices with ∞ entries. For a successively solvable shift matrix A with only finite entries, there must exists a path P from $\{A\}$ that includes a cycle. For any subset $\mathcal{C}$ of entries (i, j) in A such that R_{ij} is not used by any edge in the path P, let A' be the matrix obtained by modifying these entries of A in $\mathcal{C}$ to be ∞. The sequence of reducing operations used in P can be performed on A' successively, and hence it can be argued that A' is also successively solvable.

However, our understanding to shift matrices is not complete. There are more research to be done for shift matrices with ∞ entries. Further directions of research also include the characterization of (extended) cycle components and algorithms for searching cycles in $\mathcal{G}_n$.

References

[1] C. W. Sung and X. Gong, "A ZigZag-decodable code with the MDS property for distributed storage systems," in *IEEE Int. Symp. on Inf. Theory*, Jul. 2013, pp. 341–345.

[2] X. Fu, Z. Xiao, and S. Yang, "Overhead-free in-place recovery scheme for XOR-based storage codes," in *IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications*, Sep. 2014, pp. 552–557.

[3] M. Dai, X. Wang, H. Wang, X. Lin, and B. Chen, "Bandwidth overhead-free data reconstruction scheme for distributed storage code with low decoding complexity," *IEEE Access*, vol. 5, pp. 6824–6832, 2017.

[4] X. Gong and C. W. Sung, "Zigzag decodable codes: Linear-time erasure codes with applications to data storage," *J. Comput. Syst. Sci.*, vol. 89, pp. 190–208, 2017.

[5] M. Dai, C. W. Sung, H. Wang, X. Gong, and Z. Lu, "A new Zigzag-decodable code with efficient repair in wireless distributed storage," *IEEE Trans. Mob. Comput.*, vol. 16, no. 5, pp. 1218–1230, 2017.

[6] M. Dai, B. Mao, X. Gong, C. W. Sung, W. Zhuang, and X. Lin, "Zigzag-division multiple access for wireless networks with long and heterogeneous delays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2822–2835, 2019.

[7] H. Hou, K. W. Shum, M. Chen, and H. Li, “BASIC regenerating code: Binary addition and shift for exact repair,” in *IEEE Int. Symp. on Inf. Theory*, Jul. 2013, pp. 1621–1625.
[8] X. Fu, Z. Xiao, and S. Yang, “Overhead-free in-place recovery and repair schemes of XOR-based regenerating codes,” in *IEEE Int. Symp. on Inf. Theory*, Jul. 2015, pp. 341–345.
[9] H. Hou, P. P. C. Lee, and Y. S. Han, “Zigzag-decodable reconstruction codes with asymptotically optimal repair for all nodes,” *IEEE Transactions on Communications*, vol. 18, no. 10, pp. 5999–6011, Oct. 2020.
[10] X. Fu, S. Yang, and Z. Xiao, “Decoding and repair schemes for shift-XOR regenerating codes,” *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7371–7386, 2020.
[11] T. Nozaki, “Fountain codes based on zigzag decodable coding,” in *Proc. IEEE Int. Symp. on Information Theory and its Applications*, 2014, pp. 274–278.
[12] B. Jun, P. Yang, J. No, and H. Park, “New fountain codes with improved intermediate recovery based on batched zigzag coding,” *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 23–36, 2017.
[13] Y. Murayama and T. Nozaki, “Efficient scheduling of serial iterative decoding for zigzag decodable fountain codes,” in *International Symposium on Information Theory and Its Applications, ISITA 2018, Singapore, October 28-31, 2018*. IEEE, 2018, pp. 286–290.
[14] P. Shi, Z. Wang, D. Li, and W. Xiang, “Zigzag decodable online fountain codes with high intermediate symbol recovery rates,” *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6629–6641, 2020.
[15] C. W. Sung and X. Gong, “Combination network coding: Alphabet size and zigzag decoding,” in *2014 International Symposium on Information Theory and its Applications*. IEEE, 2014, pp. 699–703.
[16] M. Luby, “LT codes,” in *Proc. of the 43rd Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, Nov. 2002, pp. 271–280.
[17] A. Shokrollahi, “Raptor codes,” *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
[18] S. Gollakota and D. Katabi, “Zigzag decoding: combating hidden terminals in wireless networks,” in *ACM SIGCOMM*, vol. 38, no. 4, Aug. 2008, pp. 159–170.
[19] M. Dai, B. Mao, X. Gong, C. W. Sung, W. Zhuang, and X. Lin, “Zigzag-division multiple access for wireless networks with long and heterogeneous delays,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2822–2835, 2019.
[20] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Cambridge, England, 2008.
[21] X. Cheng, X. Fu, Y. Guo, K. W. Shum, and S. Yang, “Successively solvable shift-add systems — a graphical characterization,” full version. [Online]. Available: https://shhyang.github.io/research/shift-add-full.pdf