# The Positive Capacity Region of Two-Dimensional Run Length Constrained Channels

Keren Censor
Department of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: ckeren@cs.technion.ac.il

Tuvi Etzion
Department of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: etzion@cs.technion.ac.il

***Abstract*— A binary sequence satisfies a one-dimensional $(d,k)$ constraint if every run of zeroes has length at least $d$ and at most $k$. A binary two-dimensional array satisfies a $(d,k)$ constraint if every run of zeroes, in each one of the array directions, has length at least $d$ and at most $k$. Few models have been proposed in the literature to handle two dimensional data: the diamond model, the square model, the hexagonal model, and the triangular model. The constraints in the different directions might be asymmetric and hence many kind of constraints are defined depending on the number of directions in the model. For example, a two-dimensional array in the diamond model satisfies a $(d_1, k_1, d_2, k_2)$ constraint if it satisfies the one-dimensional $(d_1, k_1)$ constraint horizontally and the one-dimensional $(d_2, k_2)$ constraint vertically. In this paper we examine the region in which the capacity of the constraints is zero or positive in the various models. We consider asymmetric constraints in the diamond model and symmetric constraints in the other models. In particular we provide an almost complete solution for asymmetric constraints in the diamond model.**

## I. Introduction

Runlength constraint coding is widely used in digital storage applications, particularly magnetic and optical storage devices [3], [4]. Recent developments in optical storage – especially in the area of holographic memory – increase recording density by exploiting the fact that the recording device is a surface. In this new model, the recording data is regarded as two-dimensional, as opposed to the track-oriented one-dimensional recording paradigm. This new approach, however, necessitates the introduction of new types of constraints which are two-dimensional rather than one-dimensional. While the one-dimensional case has been widely explored, results in the two-dimensional case have been slower to arrive. This is mainly due to the fact that imposing constraints in a few directions makes the coding problem much more difficult.

A one-dimensional binary sequence is said to satisfy a $(d,k)$ constraint if there are at least $d$ *zeroes* and at most $k$ *zeroes* between any pair of consecutive *ones*. A two-dimensional surface is said to satisfy a $(d,k)$ constraint if each direction defined by its connectivity model satisfies a one-dimensional $(d,k)$ constraint (with possibly runs smaller than $d$ on the edges of the array). The two-dimensional capacity of a two-dimensional constraint $\Theta$ is defined by

$$C(\Theta) = \lim_{n,m\to\infty} \frac{\log_2 N(n,m \mid \Theta)}{rnm},$$

where $N(n,m \mid \Theta)$ is the number of $n \times m$ arrays satisfying the constraint $\Theta$ and $rnm$ is the number of points in an $n \times m$ array for the given connectivity model. An array which satisfies the constraint $\Theta$ is called $\Theta$ *constrained* or $\Theta$ *array*.

Data should be organized on a two-dimensional surface in some order which defines the way in which the data is read. For this purpose four connectivity models are defined. The diamond model, the square model, and the hexagonal model, for constrained codes were considered by Weeks and Blahut [9], while the triangular model was considered by [8] for constrained codes and other applications in [2].

The first connectivity model is the *diamond* model. In this model, a point $(i,j) \in \mathbb{Z}^2$ has the following four neighbors:

$$\{(i+1,j),(i-1,j),(i,j+1),(i,j-1)\}.$$

When $(i,j)$ is an edge point, the neighbor set is reduced to points within the array. In this model the data is organized in the two-dimensional rectangular grid and it is read horizontally and vertically.

The second model is called the *square* model, in which each point $(i,j) \in \mathbb{Z}^2$ has eight neighbors:

$$\{(i+1,j),(i-1,j),(i,j+1),(i,j-1),\\ (i+1,j+1),(i-1,j+1),(i+1,j-1),(i-1,j-1)\}.$$

In this model the data is organized in the two-dimensional rectangular grid and it is read horizontally, vertically, and in the two diagonal directions.

The third model is called the *hexagonal* model. Instead of the rectangular grid, we define the following graph. We start by tiling the plane $\mathbb{R}^2$ with regular hexagons. The vertices of the graph are the center points of the hexagons. These points define the hexagonal lattice. We connect two vertices if and only if their respective hexagons are adjacent.

We will use an isomorphic representation of the model. This representation includes $\mathbb{Z}^2$ as the set of vertices. Each point $(i,j) \in \mathbb{Z}^2$ has the following neighboring vertices,

$$\{(i+1,j),(i-1,j),(i,j+1),\\ (i,j-1),(i-1,j-1),(i+1,j+1)\}.$$

The two models are isomorphic. From now on, by abuse of notation, we will also call the last model – the hexagonal model. In this isomorphic model the data is organized in the two-dimensional rectangular grid and it is read horizontally, vertically, and in one of the diagonals direction, called *right diagonal*. In all models rows and columns of the arrays will be indexed in ascending order, bottom to top and left to right.

The fourth model is called the *triangular* model. Again, we start by tiling the plane $\mathbb{R}^2$ with regular hexagons. The vertices of the graph are the vertices of the hexagons. The edges between the vertices are the sides of the hexagons. Hence, each vertex has exactly three neighboring vertices. If we connect the centers of the hexagons with lines we will obtain a tiling of the $\mathbb{R}^2$ with equilateral triangles. The vertices of the graph are union of two translates of the hexagonal lattice. Clearly, a point in this model can be represented by a triple $(i, j, s) \in \mathbb{Z}^2 \times \{0, 1\}$. Each point $(i, j, 0) \in \mathbb{Z}^2 \times \{0\}$ has the following neighboring vertices

$$\{(i, j, 1), (i-1, j, 1), (i, j-1, 1)\}.$$

Each point $(i, j, 1) \in \mathbb{Z}^2 \times \{1\}$ has the following neighbors

$$\{(i, j, 0), (i+1, j, 0), (i, j+1, 0)\}.$$

As the vertices are two translates of the hexagonal lattice, one can consider the model as having six directions. We will consider it slightly different. Instead of data stored in the centers of the triangles, the data will occupy the whole area of the triangle. Therefore, there are three directions in this model and an $n \times m$ array has $2nm$ points (see Fig. 10).

Let $C_\diamond(d, k)$ denote the capacity of the $(d, k)$ two-dimensional constraint in the diamond model. Kato and Zeger [5] proved that $C_{\diamond(d,k)} > 0$ if and only if $k > d+1$. $C_\diamond(d_1, k_1, d_2, k_2)$ denotes the capacity of the asymmetric $(d_1, k_1, d_2, k_2)$ constraint in the diamond model [6], i.e., horizontally the constraint is $(d_1, k_1)$ and vertically the constraint is $(d_2, k_2)$. $C_\boxplus(d, k)$, $C_\bigcirc(d, k)$, $C_\triangle(d, k)$, denote the capacity of the $(d, k)$ constraint in the square model, hexagonal model, and triangular model, respectively.

The rest of this paper is organized as follows. In Section II basic techniques to prove zero or positive capacity are presented. In Section III we provide an almost complete solution for the zero/positive capacity region problem for asymmetric constraints in the diamond model. In Sections IV, V, and VI we examine capacities of constraints in the square model, hexagonal model, and triangular model, respectively.

## II. Basic Techniques

In this section we will present known techniques, used to prove zero capacity and those used to prove positive capacity. The first lemma which appeared in [6] is an immediate consequence from the definition of the $(d, k)$ constraint.

*Lemma 1:* Let $\Theta$ be a constraint with minimum runlength $d$ and maximum runlength $k$ in direction $\Delta$. Let $\tilde{\Theta}$ be a constraint with minimum runlength $\tilde{d} \leq d$ and maximum runlength $\tilde{k} \geq k$ in direction $\Delta$ and the same constraints in the other directions. Then $C(\Theta) \leq C(\tilde{\Theta})$.

### A. Positive Capacity

An $[n \times m, k \times \ell]$ *skeleton tile* is a tile which consists of an $n \times m$ array from which an $k \times \ell$ array was removed from the upper right corner. If $\ell = 1$ we simply have an $[n \times m, k]$ skeleton tile. For two points $z_1 = (x_1, y_1)$ and $z_2 = (x_2, y_2)$, $z_1, z_2 \in \mathbb{Z}^2$ let $\mathcal{L}(z_1, z_2) = \{(ix_1 + jx_2, iy_1 + jy_2) : i, j \in \mathbb{Z}\}$ the *set of points spanned by* $z_1, z_2$. This is the lattice defined by $z_1$ and $z_2$. Note, that by abuse of notation the first coordinate is for the row index and the second is for the column index. The following lemma can be easily verified.

*Lemma 2:* Let $\mathcal{A}$ be an $[n \times m, k \times \ell]$ skeleton tile. If we place the bottom leftmost point of $\mathcal{A}$ on the points of $\mathcal{L}((n-k, m-\ell), (n, -\ell))$ then we will obtain a tiling of $\mathbb{R}^2$ with copies of $\mathcal{A}$.

The tiling obtained by Lemma 2 will be called the *standard tiling*. If $\mathcal{A}$ is an $n \times m$ array (a *skeleton array*) then the standard tiling is obtained by substituting $k = 0$ and $\ell = 0$ in the skeleton tile of lemma 2. A standard tiling can use a few tiles with the same shape and different labels. In this case each one of the tiles can have any one of the labels. The next lemma is a straight forward generalization of a similar lemma for skeleton arrays, given in [6].

*Lemma 3:* Let $\mathcal{A}$ and $\mathcal{B}$ be two different labels of the same tile and $\Theta$ a two-dimensional constraint. If the standard tiling with $\mathcal{A}$ and $\mathcal{B}$ yields a two-dimensional array which is $\Theta$ constrained then $C(\Theta) > 0$.

### B. Zero Capacity

Blackburn [1] gave a method to prove zero capacity for specific constraints on both *zeroes* and *ones*. But, the method can be formulated to handle general two-dimensional constraints.

Assume we want to show that the capacity of a two-dimensional constraint $\Theta$ is zero. We consider an $(n + r_1 + r_2) \times (m + t_1 + t_2)$ array $\mathcal{A}$ which is $\Theta$ constrained, where $t_1$, $t_2$, $r_1$, and $r_2$ are constants which might depend on the runlength constraints, but do not depend on $n$ and $m$. Assume further that the labels at positions of the first $r_1$ rows, the last $r_2$ rows, the first $t_1$ columns, and the last $t_2$ columns, are known. We now scan the other positions of $\mathcal{A}$. We scan the other $n$ rows from bottom to top, and the $m$ positions in a row are scanned from left to right. If each position is determined by the known labels and the positions which are already scanned then the capacity of the constraint $\Theta$ is *zero*. We will call this technique *scanning*. The strength of scanning is demonstrated by providing a very short proof to the following theorem by Kato and Zeger [5].

*Theorem 1:* $C_\diamond(d, d+1) = 0$

*Proof:* Consider an $n \times m$ array $\mathcal{A}$ which is $(d, d+1)$ constrained. We will show that the labels of $\mathcal{A}$ are determined by the labels at positions $(i, j)$, where $0 \leq i \leq d$ or $0 \leq j \leq d-1$ or $j = m-1$.

We will show that for every $d+1 \leq i$, $d \leq j \leq m-2$, the label $X$ at position $(i, j)$ is determined by labels to the left of it and labels below it (see Fig. 1). Assume the contrary that $X$ can be a *zero* and can be a *one*. It implies that all the positions marked by $A$ are *zeros* and either $X$ or $Y$ is a *one*.

Since $Y$ can be a *one*, it follows that all positions marked by $B$ are labelled by *zeroes*. Since $X$ can be a *zero* it follows by the vertical constraint that $C$ is a *one*. Similarly, since $Y$ can be a *zero*, it follows that $D$ is a *one*, a contradiction to the horizontal constraint. Hence, $C_\diamond(d, d+1) = 0$.

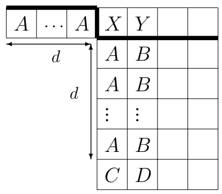


Fig. 1. Scanning of a $(d, d+1)$ array.

■

The technique is generalized as follows.

*Theorem 2:* Assume the scanning method is applied to the two-dimensional constraint $\Theta$, and in each position $(i, j)$ scanned one of the following three states holds:

(s1) The label in position $(i, j)$ is completely determined.
(s2) The label in position $(i, j)$ can be either *zero* or *one*, but with one of these labels the suffix of the row is completely determined.
(s3) The label in position $(i, j)$ can be either *zero* or *one*, but the prefix of the row before position $(i, j)$ is a given sequence $\mathcal{P}(i, j)$ .

Then $C(\Theta) = 0$.

The theorem is proved by showing that if the number of positions to be labelled in a row is $r$ then there are at most $\frac{(r+2)(r-1)}{2}$ different ways to label the row.

## III. Asymmetric Run-length Constrained Channels

Kato and Zeger [6] have considered the zero/positive region of $C_\diamond(d_1, k_1, d_2, k_2)$. They have summarized their results in which seven cases remained unsolved:

(u1) $d_1 = 1,\ k_1 = 3,\ d_2 = 2,\ k_2 = 3$.
(u2) $2 \le d_1,\ k_1 = d_1 + 1,\ d_2 = d_1,\ k_2 \le 2d_2$.
(u3) $2 \le d_1,\ d_1 + 2 \le k_1 \le 2d_1,\ d_2 = d_1,\ k_2 = d_2 + 1$.
(u4) $2 \le d_1 < d_2 < k_1 - 1,\ d_1 + 2 \le k_1 \le 2d_1,\ k_2 = d_2 + 1$.
(u5) $2 \le d_1,\ d_1 + 2 \le k_1 \le 2d_1,\ d_2 = k_1 - 1,\ k_2 \le 2d_2$.
(u6) $2 \le d_1,\ 2d_1 < k_1,\ d_1 < d_2 < k_1 - 1,\ k_2 = d_2 + 1$.
(u7) $2 \le d_1,\ 2d_1 < k_1,\ d_2 = k_1 - 1,\ k_2 \le 2d_2$.

*Lemma 4:* $C_\diamond(d, 2d+1, 2d, 2d+1) > 0$ for every $d \ge 1$.

*Proof:* Let $T_n$ be a $(2n-2) \times (2n)$ array defined as follows. $T_n(1, 2n-2) = 1$ and $T_n(0, n-2) = 1$; if $T_n(i, j) = 1$ then $T_n(i+2, j-1) = 1$ provided that $i + 2 \le 2n - 3$. In all other positions $T_n$ has $zeroes$ (see Fig. 2).

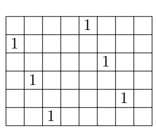

Fig. 2. The array $T_4$.

Consider the $[(4d+4) \times (2d+3), 2d+3]$ skeleton tile of Fig. 3. Let $\mathcal{A}$ and $\mathcal{B}$ be the two $[(4d+4) \times (2d+3), 2d+3]$ tiles obtained from the skeleton tile by substituting the two skew tetrominoes of Fig. 4 instead of the four asterisks. The standard tiling with the arrays $\mathcal{A}$ and $\mathcal{B}$ yields a $(d, 2d+1, 2d, 2d+1)$ constrained array. Therefore, by Lemma 3 $C_\diamond(d, 2d+1, 2d, 2d+1) > 0$.

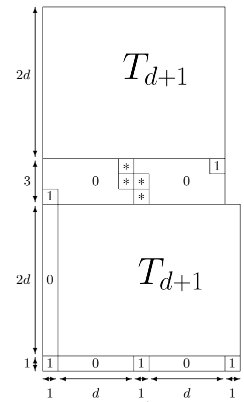


Fig. 3. The skeleton tile for the $(d, 2d+1, 2d, 2d+1)$ constraint.

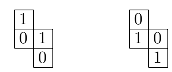

Fig. 4. Two skew tetrominoes for substitution in the skeleton tile.

■

*Lemma 5:* $C_\diamond(d, 2d+2, 2d+1, 2d+2) > 0$ for every $d \ge 1$.

*Proof:* Consider the $(4d+5) \times (2d+3)$ skeleton array of Fig. 5. Let $\mathcal{A}$ and $\mathcal{B}$ be the two arrays obtained from the skeleton array by substituting a $one$ instead one of the asterisks and a $zero$ instead of the second. The standard tiling with $\mathcal{A}$ and $\mathcal{B}$ yields a $(d, 2d+2, 2d+1, 2d+2)$ constrained array. Thus, by Lemma 3 $C_\diamond(d, 2d+2, 2d+1, 2d+2) > 0$.

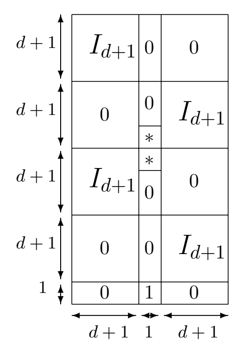


Fig. 5. A skeleton array for the $(d, 2d+2, 2d+1, 2d+2)$ constraint.

■

*Lemma 6:* If $d_1 \ge 1$, $k_1 > 2d_1$, $d_2 = k_1 - 1$, and $k_1 \le k_2 \le 2d_2$ then $C_\diamond(d_1, k_1, d_2, k_2) > 0$.

*Proof:* Assume $d_1 \ge 1$, $k_1 = 2d_1 + t$, $t > 0$, $d_2 = k_1 - 1$, and $k_2 = k_1$. We distinguish between two cases:

**Case 1:** $t = 2r + 1$, $r \ge 0$.

By Lemma 4 we have $C_\diamond(d_1 + r, 2d_1 + 2r + 1, 2d_1 + 2r, 2d_1 + 2r + 1) > 0$. Therefore, by Lemma 1 we have $C_\diamond(d_1, 2d_1 + 2r + 1, 2d_1 + 2r, 2d_1 + 2r + 1) > 0$.

**Case 2:** $t = 2r + 2$, $r \ge 0$.

By Lemma 5 we have $C_\diamond(d_1+r, 2d_1+2r+2, 2d_1+2r+1, 2d_1+2r+2) > 0$. Therefore, by Lemma 1 we have $C_\diamond(d_1, 2d_1+2r+2, 2d_1+2r+1, 2d_1+2r+2) > 0$.

Hence, $C_\diamond(d_1, 2d_1+t, 2d_1+t-1, 2d_1+t) > 0$ and thus by Lemma 1 we have that if $d_1 \geq 1$, $k_1 > 2d_1$, $d_2 = k_1 - 1$ and $k_1 \leq k_2 \leq 2d_2$ then $C_\diamond(d_1, k_1, d_2, k_2) > 0$. ■

*Lemma 7:* If $d \geq 2$ and $d-1 \geq r \geq 1$ then $C_\diamond(d, 2d+1, d+r, d+r+1) > 0$.

*Proof:* We first define a $(d+r-1) \times d$ array $H_{d,r}$ recursively as follows. For $\rho \geq 1$ let,

$$H_{\delta,2\rho} = \left( \begin{array}{cccc} & & & 0 \\ & H_{\delta-1,2\rho-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 \end{array} \right), \quad H_{\delta,2\rho+1} = \left( \begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & H_{\delta-1,2\rho} & \\ 0 & & & \end{array} \right),$$

where $H_{\delta,1} = I_\delta$. The $(d+r-1) \times d$ array $H'_{d,r}$ is defined to be the rotation of $H_{d,r}$ by $180°$.

Consider the $[(2d+2r+4) \times (3d+2), 2d+2r+1]$ skeleton tile of Fig. 6. Let $\mathcal{A}$ and $\mathcal{B}$ be the two tiles obtained from the skeleton tile by substituting the two skew tetrominoes of Fig. 4 instead of the four asterisks. The standard tiling with the arrays $\mathcal{A}$ and $\mathcal{B}$ yields a $(d, 2d+1, d+r, d+r+1)$ constrained array. Thus, by Lemma 3 $C_\diamond(d, 2d+1, d+r, d+r+1) > 0$.

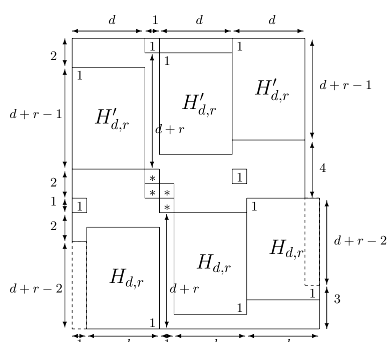


Fig. 6. The skeleton tile for $(d, 2d+1, d+r, d+r+1)$ constraint.

■

*Lemma 8:* If $d_1 \geq 2$, $k_1 > 2d_1$, $d_1 < d_2 < k_1 - 1$, and $k_2 = d_2 + 1$ then $C_\diamond(d_1, k_1, d_2, k_2) > 0$.

*Proof:* We distinguish between two cases:

**Case 1:** $d_1 < d_2 < 2d_1$.

By Lemma 7 we have $C_\diamond(d_1, 2d_1+1, d_2, d_2+1) > 0$ and hence by Lemma 1 we have $C_\diamond(d_1, k_1, d_2, d_2+1) > 0$.

**Case 2:** $2d_1 \leq d_2 < k_1 - 1$.

By Lemma 6 we have $C_\diamond(d_1, d_2+1, d_2, d_2+1) > 0$ and hence by Lemma 1 we have $C_\diamond(d_1, k_1, d_2, d_2+1) > 0$. ■

By using the scanning method we obtain the following result.

*Proposition 1:* If $d_1 \geq 2$, $k_1 \leq 2d_1$, $d_1 \leq d_2 \leq k_1 - 1$, and $k_2 = d_2 + 1$ then $C_\diamond(d_1, k_1, d_2, k_2) = 0$.

The results in this section produce solutions to most of the seven unsolved cases. (u1) is solved in Lemma 4, (u2), (u3), and (u4) in Proposition 1, (u6) in Lemma 8, and (u7) in Lemma 6. (u5) was solved when $k_2 = d_2 + 1$ in Proposition 1. The only case which remained unsolved is $2 \leq d_1$, $d_1 + 2 \leq k_1 \leq 2d_1$, $d_2 = k_1 - 1$, and $d_2 + 2 \leq k_2 \leq 2d_2$.

## IV. The Square Model

Let $\mathcal{P}$ and $\mathcal{Q}$ be the two $5 \times 5$ permutation arrays given in Fig. 7. The *ones* in both arrays occupy the same rows, columns and diagonals. Therefore we have the following lemma.

*Lemma 9:* If $\mathcal{A}$ is an $n \times n$ $(d, k)$ array then any exchanges of copies of $\mathcal{P}$ with copies of $\mathcal{Q}$ in disjoint positions of $\mathcal{A}$ will result in a $(d-3, k+3)$ constrained array.

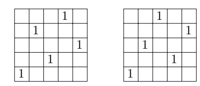

Fig. 7. Two $5 \times 5$ exchangeable arrays.

For $i \not\equiv 2 \ (mod\ 3)$ let

$$\mathcal{L}_i = \{(x, y) \ : \ x = 2j + \ell, \ y = j + i\ell, \ j, \ell \in \mathcal{Z}\}$$

be a set of points in $\mathbb{Z}^2$.

*Lemma 10:* Let $d = 2r$, $r \not\equiv 1 \ (mod\ 3)$ be an even integer and let $\mathcal{A}$ be an $n \times n$ binary array, where $A_{ij} = 1$ if $(i, j) \in \mathcal{L}_{r+1}$. Then $\mathcal{A}$ is a $(d, d)$ constrained array.

For two arrays $\mathcal{A}$ and $\mathcal{B}$ let $\mathcal{A} \times \mathcal{B}$ denote the direct product of $\mathcal{A}$ and $\mathcal{B}$.

*Lemma 11:* If $\mathcal{A}$ is a $(d, d)$ constrained array then $\mathcal{A} \times \mathcal{P}$ is a $(5d+4, 5d+4)$ constrained array.

From Lemmas 3, 9, 10 and 11 we have:

*Theorem 3:* $C_\boxplus(d, d+6) > 0$, $d \equiv 1, \ 21 \ (mod\ 30)$.

By using two similar $7 \times 7$ permutation arrays we obtain.

*Theorem 4:* $C_\boxplus(d, d+8) > 0$, $d \equiv 2, \ 30 \ (mod\ 42)$.

Some slightly smaller improvements are obtained similarly. By using the scanning method we obtain:

*Theorem 5:* $C_\boxplus(d, d+3) = 0$.

## V. The Hexagonal Model

The first result is due to Kukorelly and Zeger [7], [10]:

*Theorem 6:*

- If $d \equiv 0 \ (mod\ 6)$ then $C_\bigcirc(d, d+4) > 0$.
- $C_\bigcirc(d, d+2) = 0$.
- If $d \in \{3, 4, 5, 7, 9, 11\}$ then $C_\bigcirc(d, d+3) = 0$.

Let $\mathcal{A}$ be an $n \times n$ hexagonal array. We say that $\mathcal{A}$ has $n$ rows, $n$ columns, and $n$ right diagonals. $A_{i,j}$ belongs to row $i$, column $j$, right diagonal $[j-i]_n$, where $[\alpha]_n$ is an integer $\beta$ such that $0 \leq \beta \leq n-1$, and $\alpha \equiv \beta \ (mod\ n)$. An $n \times n$ permutation array is called *doubly periodic non-attacking semi-queens* array if each row, each column, and each right diagonal has exactly one *one*.

*Lemma 12:* A standard tiling of a two-dimensional array with a $(d+1) \times (d+1)$ doubly periodic non-attacking semi-queens array will result in a $(d, d)$ constrained array.

*Lemma 13:* If $n$ is even then there is no doubly periodic $n \times n$ non-attacking semi-queens array.

For even $n \geq 6$, $(n+3) \times (n+3)$ doubly periodic non-attacking semi-queens arrays exist for all $n$'s. We use the following $(n+3) \times (n+3)$ skeleton array:

$$\mathcal{B} = \left[ \begin{array}{cc} \mathbf{0} & \mathcal{P} \\ H_n & \mathbf{0} \end{array} \right],$$

where $H_n$ is an appropriate $n \times n$ permutation array, and $\mathcal{P}$ is a $3 \times 3$ array. Let $\mathcal{A}_{n+3}$ and $\mathcal{B}_{n+3}$ be the two $(n+3) \times (n+3)$ arrays obtained from the skeleton array by substituting in $\mathcal{P}$ the two $3 \times 3$ arrays shown in Fig. 8. If $\mathcal{A}_{n+3}$ and $\mathcal{B}_{n+3}$ are $(n+3) \times (n+3)$ doubly periodic non-attacking semi-queens arrays then we will have that $C_{\bigcirc}(n, n+4) > 0$.

Fig. 8. Two $3 \times 3$ exchangeable arrays

In the construction we distinguish between the even values of $n$ modulo 10. Each such value has a different construction. The first two constrained arrays are presented in Fig. 9. Hence we have the following theorem:

*Theorem 7:* $C_{\bigcirc}(d, d+4) > 0$, for even $d > 5$.

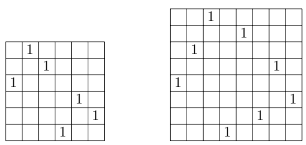

Fig. 9. $H_6$ and $H_8$

## VI. The Triangular Model

Let $\mathcal{A}$ be an $n \times n$ triangular array. We say that $\mathcal{A}$ has $n$ rows, $n$ right columns, and $n$ left columns. $A_{i,j,s}$ belongs to row $i$, right column $j$, left column $[i+j+s]_n$ (see Fig. 10).

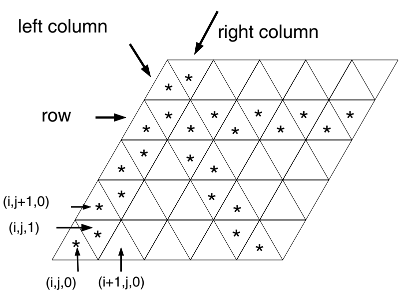


Fig. 10. A $5 \times 5$ triangular array

An $n \times n$ triangular array is called *doubly periodic non-attacking triangle queens* array if each row, each right column, and each left column has exactly one *one*. For even $n$, let $\mathcal{T}_n$ be an $n \times n$ triangular array defined by $\mathcal{T}_n(i, i, s) = 1$ if $s \not\equiv i \ (mod\ 2)$, $0 \leq i \leq n-1$. All other positions of $\mathcal{T}_n$ are zeroes. $\mathcal{T}_6$ is illustrated in Fig. 11.

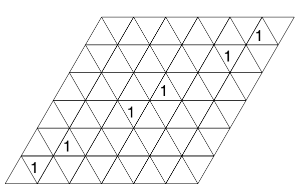

Fig. 11. The array $\mathcal{T}_6$.

$\mathcal{T}_n$ is an $n \times n$ doubly periodic non-attacking triangle queens array. The standard tiling with $\mathcal{T}_n$ is an $(2n-1, 2n-1)$ constrained array. Any exchanges in this tiling of the leftmost $2 \times 2$ triangular array of Fig. 12, with any of the other two triangular arrays of Fig. 12, will result in an $(2n-3, 2n+1)$ array. Hence, we have:

*Theorem 8:* If $d \equiv 1 \ (mod\ 4)$ then $C_{\triangle}(d, d+4) > 0$.

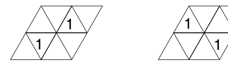

Fig. 12. Three $2 \times 2$ exchangeable triangular arrays.

For $d \equiv 3 \ (mod\ 4)$ a similar construction cannot work.

The proof for zero capacity will be heavily based on the existence of two patterns, Podd and Peven, in the constrained arrays. These patterns are depicted in Figures 13 and 14.

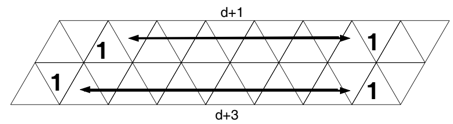


Fig. 13. The pattern Podd

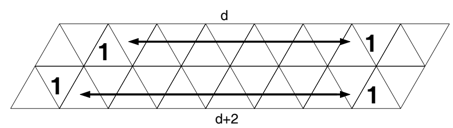


Fig. 14. The pattern Peven.

*Lemma 14:* Let $d \geq 5$ be an odd (even) integer, $h = \frac{d+7}{2}$ ($h = \frac{d+6}{2}$), and let $\mathcal{A}$ be a $(d, d+3)$ infinite array. If $\mathcal{A}$ contains an $r \times h$ subarray $\mathcal{B}$ whose first two rows form the pattern Podd (Peven), then the first two and the last two right columns of $\mathcal{B}$ are substrings of $(10^{d+2})^\infty$ $((10^{d+1})^\infty)$.

Now, we use scanning and make use of Lemma 14 and Theorem 2 to prove the following theorem.

*Theorem 9:* $C_{\triangle}(d, d+3) = 0$ if $d \geq 3$.

## References

[1] S. R. Blackburn, "Two dimensional runlength constrained arrays with equal horizontal and vertical constraints ," *IEEE Trans. on Inform. Theory*, submitted .

[2] S. I. Costa, M. Muniz, E. Agustini, R. Palazzo, "Graphs, tessellations, and Perfect codes on flat tori," *IEEE Trans. on Inform. Theory*, vol. IT-50, pp. 2363-2377, October 2004.

[3] K. A. S. Immink, *Coding Techniques for Digital Recorders*, New York: Prentice-Hall, 1991.

[4] K. A. S. Immink, *Codes for Mass Data Storage Systems*, The Netherlands: Shannon Foundation Publishers, 1999.

[5] A. Kato and K. Zeger, "On the capacity of two-dimensional run length constarined channels," *IEEE Trans. on Inform. Theory*, vol. 45, pp. 1527-1540, July 1999.

[6] A. Kato and K. Zeger, "Partial characterization of the positive capacity region of two-dimensional asymmetric run length constrained channels ," *IEEE Trans. on Inform. Theory*, vol. 46, pp. 2666-2670, November 2000.

[7] Zs. Kukorelly and K. Zeger, "The capacity of some hexagonal (d,k) constraints ," *IEEE International Symposium on Information Theory*, p. 263, Washington, Dc, June 2001.

[8] Zs. Nagy and K. Zeger, "Capacity bounds for the hard-triangle modes," *IEEE International Symposium on Information Theory*, p. 162, Chicago, Illinois, June 2004.

[9] W. Weeks and R. E. Blahut, "The capacity and coding gain of certain checkerboard codes," *IEEE Trans. on Inform. Theory*, vol. 44, pp. 1193–1203, May 1998.

[10] K. Zeger, personal communication.