

Article

# A Vertex-Aligned Model for Packing 4-Hexagonal Clusters in a Regular Hexagonal Container

Marina Prvan <sup>1,\*</sup> , Arijana Burazin Mišura <sup>2</sup>, Zoltan Gecse <sup>3</sup> and Julije Ožegović <sup>1</sup>

<sup>1</sup> Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (FESB), University of Split, 21000 Split, Croatia; julije@fesb.hr

<sup>2</sup> University Department of Professional Studies, University of Split, 21000 Split, Croatia; aburazin@oss.unist.hr

<sup>3</sup> Fermi National Accelerator Laboratory (Fermilab), Batavia, IL 60510-5011, USA; zgecse@cern.ch

\* Correspondence: mprvan@fesb.hr

Received: 30 January 2020; Accepted: 28 February 2020; Published: 1 May 2020



**Abstract:** This paper deals with a problem the packing polyhex clusters in a regular hexagonal container. It is a common problem in many applications with various cluster shapes used, but symmetric polyhex is the most useful in engineering due to its geometrical properties. Hence, we concentrate on mathematical modeling in such an application, where using the “bee” tetrahex is chosen for the new Compact Muon Solenoid (CMS) design upgrade, which is one of four detectors used in Large Hadron Collider (LHC) experiment at European Laboratory for Particle Physics (CERN). We start from the existing hexagonal containers with hexagonal cells packed inside, and uniform clustering applied. We compare the center-aligned (CA) and vertex-aligned (VA) models, analyzing cluster rotations providing the increased packing efficiency. We formally describe the geometrical properties of clustering approaches and show that cluster sharing is inevitable at the container border with uniform clustering. In addition, we propose a new vertex-aligned model decreasing the number of shared clusters in the uniform scenario, but with a smaller number of clusters contained inside the container. Also, we describe a non-uniform tetrahex cluster packing scheme in the proposed container model. With the proposed cluster packing solution, it is accomplished that all clusters are contained inside the container region. Since cluster-sharing is completely avoided at the container border, the maximal packing efficiency is obtained compared to the existing models.

**Keywords:** cluster packing; polyhex; regular hexagon; tetrahex

## 1. Introduction

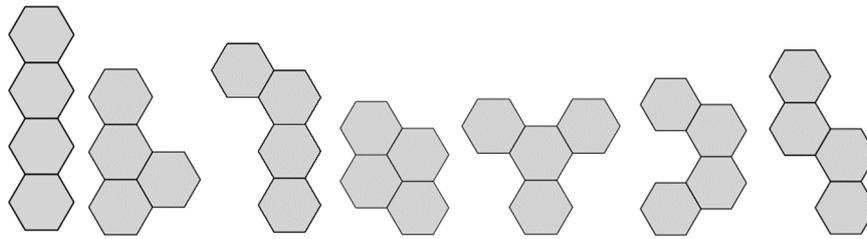
A problem of packing objects in the container of a given shape is common in many applications such as computer science, manufacturing, industrial engineering, and production [1]. Various container types are studied, such as rectangular [2,3], circular [4], or polygonal geometrical shapes [5,6]. Also, problems of embedding many different items are investigated, where the inner components can be regular circular-like [7] or irregular objects [8,9]. Even though most of the research supports the approximation of different container shapes with the minor change of the cost function, special attention is devoted to packing polygons into the circular or polygonal region of interest (ROI) [2,10]. It is crucial in applications such as sensor manufacturing when polygonal sensors are cut out from the circular wafer [11]. In some cases, inner objects are not independent, in sense, they have to be grouped in a number of clusters before being embedded inside the container. For example, when polygonal clusters are formed inside the polygonal area covered with wireless sensor devices to obtain increased energy efficiency [12].

Clusters can be packed in the container with various objectives. For example, a cluster layout can be optimized to increase packing density with maximal distance among clusters [1], or the aim could be to minimize the size of the container still containing the cluster collection [9]. Our work is similar to studies that intend to maximize the packed area by embedding as many clusters as possible in the container area [2,3,10]. Litvinchev et al. [2] studied this problem with the aim of packing a certain number of circular-like objects such as circles, ellipses, rhombuses, and polygons of known size into a rectangular shape. Authors approximate containers by regular grids used to simplify the packing problem. This strategy was introduced earlier by Stoyan and Yaskov [13], who applied a hexagonal lattice for packing circles by assigning their centers to the nodes of the grid. Similarly, Galiev and Lisafina [10,14], developed their specific grid-based models for packing a maximal number of equal and unequal regular objects inside the square or rectangular container. Toledo et al. [15] showed that using a square grid in the packing problem can simplify even the non-trivial handling of the geometry required in case of irregular objects. However, we are interested only in regular packing, which involves standard shapes of objects and containers, unlike the ones where the distance of objects to the container centroid is not Euclidean [16,17]. In the work of Torres-Escobar et al. [3], the grid is discretized with a set of points in which circular objects can be assigned such that there are no overlaps and to maximize the space occupied. The optimal packing of discs in regular rings of uniform circles with minimal distance is examined in [18]. There are additional researches that utilize the regular grid of points, but with another packing objective, i.e., to place the fixed-size objects in a container of minimal size and area [6,19].

In this paper, we concentrate on the goal to pack as many items as possible in the regular hexagonal container and to determine the maximal number of inner packed clusters. The issue of shared clusters is only tackled in [13], where authors refer to these as prohibited areas which cause the modification of the hexagonal lattice packing. We believe that these shared or partial objects should be discussed, as they cannot be considered the inner packed items. They are unpacked, but they take place in the container and prevent us from putting the whole cluster inside the container border. They also cause increased void areas that we want to minimize and maximize the occupied space.

Unlike previous studies where not necessarily the same size of all packed objects is considered, we focus on the more uniform approach, considering all items to be the same in size and type. Also, the orthogonally oriented objects are mostly used in former research where objects are identical [2,7,13,14,16]. If the non-identical items in size and type are given, they are usually free of any orientation restrictions. Authors in [1,9], for instance, allow free rotations of the polygons and use separation lines with direct trigonometry in Cartesian coordinates to ensure non-overlapping, while Lopez and Beasley [17] separate two packing cases with the rotation of objects allowed or not. We consider the packed items to all have the same orientation but in another context. Naturally, as pointed out in [4], given the grid approximation of the container, this approach can only handle the packing of uniformly sized and uniformly oriented items inside a container. Since 4-hexagonal clusters or polyhex structures are formed by merging hexagonal cells together [20,21], we can examine several fixed and uniform cluster rotations and obtain the optimal packing solution.

We adopt the idea where the nodes of the regular hexagonal grid are considered as positions for assigning centers of the hexagonal cells. Similar to Lintzmayer [19], we utilize the hexagon tessellation method to embed hexagonal cells inside hexagonal ROI. Hexagons are chosen due to their advantages and nice properties in covering the entire plane [22,23]. We group the inner hexagonal cells into clusters of four or tetrahexes, to obtain the desired packing result of the polyhex clusters. The process of forming polyhex clusters is given in Figure 1. Since the shape of the cluster depends on the way that the cells are merged [20,21], there are seven possible combinations for clustering four hexagons, forming in total seven different 4-hexagonal clusters.



**Figure 1.** Forming the 4-hexagonal clusters: bar, pistol, worm, bee, propeller, arch, wave (Adjusted from [20,21]).

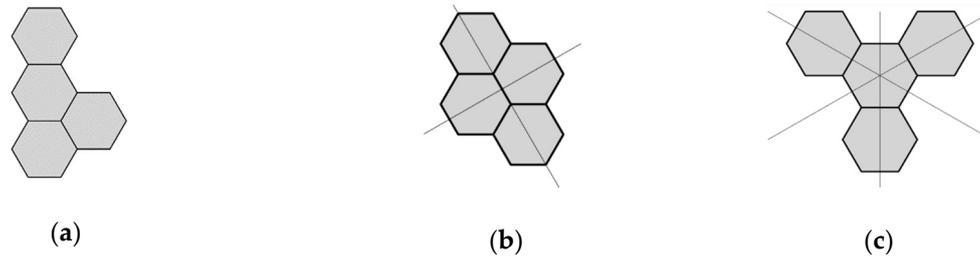
The research in this paper is application-driven, and a specific engineering application in our focus is the design upgrade of the new Compact Muon Solenoid (CMS), which is one of four detectors used in Large Hadron Collider (LHC) experiment at European Laboratory for Particle Physics (CERN). Namely, the future high granularity calorimeter (HGCAL) of CMS will consist of about 6 million channels, and not all sensor data could be read and stored for further processing. There are 40 million events per second, while only a few hundreds of events per second can currently be recorded offline. CMS uses a trigger system to select the events of interest whose data is stored and analyzed [24]. Also, the bandwidth requirements should be met in order to send the data, so a reduction is applied in several forms. One of the approaches is based on the detector geometry design, where data reduction is performed by using the clustering procedure. Hence, detector hexagonal sensor cells are grouped into larger 4-hexagonal clusters called “trigger cells”. For this purpose, hexagonal sensor modules are produced and used to cover a detector surface, where each of them contains a set of small hexagonal sensor energy readout cells grouped in 4-hexagonal clusters [25,26].

Since clusters or “trigger cells” are formed from hexagonal sensors embedded in the hexagonal HGCAL sensor module, intuitively, clusters should also be packed in the module. Hence, the mathematical problem in the current engineering context is a problem of packing 4-hexagonal (tetrahex) rhomboid clusters in the regular hexagonal module (container). The real applicability of the study is how to solve the packing problem for this specific application. Each container should contain its own clusters or, at least, the number of shared clusters at the container border should be minimized. This is to reduce or possibly to avoid communication between boards that are processing data from each of the neighboring containers separately. Ideally, the cluster plane should remain uniform to keep the simplicity of the nearest neighbor (NN) algorithm [24].

For our research, we select the densest 4-hexagonal clusters that are the most closely packed, so only three out of seven combinations could be considered (Figure 2). Since symmetry in forming “trigger cells” is of great interest for us, we choose clusters with two symmetry axes, approximated by a rhomboid shape. Furthermore, we are interested in clusters that allow period tessellation with two translation symmetries in two non-parallel directions. An additional requirement for cluster selection is the perimeter minimization. According to [27], a polyhex is optimal if it has a minimal perimeter with respect to all polyhexes with the same number of hexagons. The 4-hexagonal cluster named “bee” tetrahex allows periodic tessellation, and it is minimal in perimeter and rhomboid in shape. We find it optimal for our application, and many authors use it due to its broad applicability in engineering based on its geometry and symmetrical structure [21].

To the best of our knowledge, the problem of packing polyhex clusters composed of the hexagonal cells inside a regular hexagonal container is addressed for the first time in this paper. The main contributions are the following: First, the cluster layout for 4-hexagonal clusters packed in the regular hexagonal container is visualized and analyzed. Then, we investigate two different existing layouts: the centroid-aligned (CA) and vertex-aligned (VA) models [12,28]. In CA, the container center overlaps with the center of the central hexagonal cell, while in VA, the container center is at the vertex that belongs to one of the embedded hexagons. Next, mathematical expressions are derived, providing the number of full clusters packed in the container, as well as the number of shared clusters at the container border. Finally, we propose a new VA model (adjusted from [29]) to solve the packing problem such

that all hexagonal clusters are contained in the inner part of the container, completely avoiding shared clusters at the container border.



**Figure 2.** Selected 4-hexagonal cluster candidates. (a) pistol with no symmetry axes; (b) bee with two symmetry axes; (c) propeller with three symmetry axes.

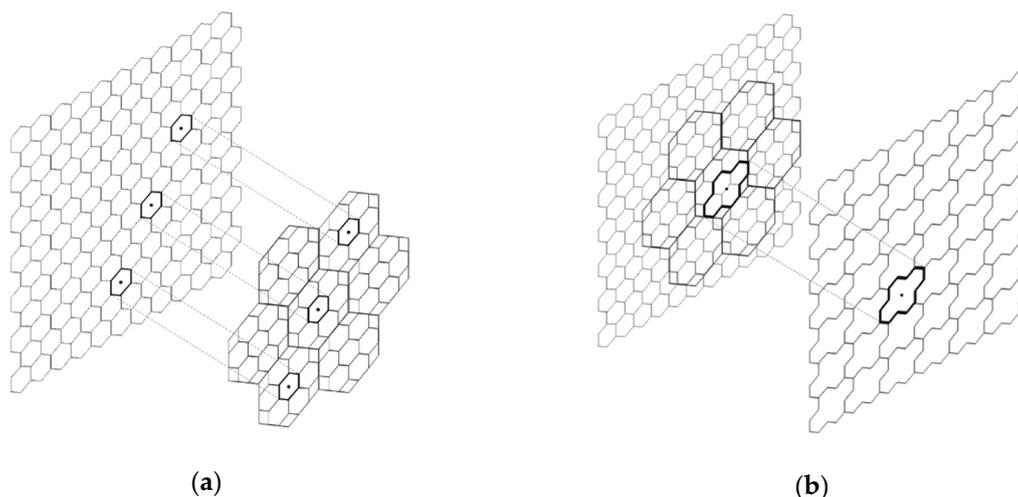
The paper is organized as follows. Section 2 provides the problem formulation and theoretical background on forming hexagonal clusters and their uniform tessellation with various cluster orientations. Next, the research methodology is formulated. Clustering with the existing models from the literature and the proposed geometrical model are presented in Sections 3 and 4 respectively. Section 5 summarizes the results with the discussion on the efficiency of architectures used for cluster packing and their comparison. Section 6 deals with the limitations of the study. The conclusion is given in Section 7 followed by the references used.

## 2. Problem Formulation

### 2.1. Formal Definition of Cluster Packing

A problem of packing polyhex clusters in the hexagonal region can be formally described as follows. First, a plane is hexagonally tessellated by using a regular hexagonal grid. Then, the same plane is covered with another regular hexagonal grid at a larger scale. These hexagons represent the regular hexagonal packing container. The size of hexagons in this new grid is defined by its (double) apothem expressed in the number of smaller hexagons (from the base tessellation).

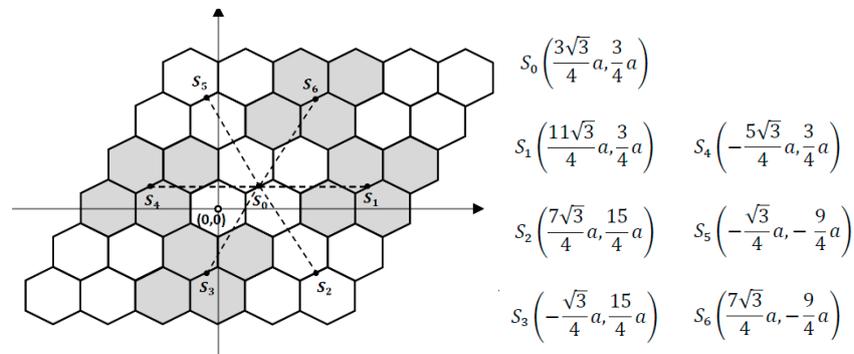
Finally, the small hexagons are packed in 4-hexagonal clusters, as shown in Figure 3. Clusters are forming a regular cluster grid or cluster plane, and the center of the defined plane must overlap with the center of one of the hexagons contained in 4-hexagonal cluster. Possibly, all clusters must be inside the container such that their packing is efficient.



**Figure 3.** A cluster packing procedure. (a) Tessellation of hexagonal containers with small hexagons inside; (b) Overlap with the regular cluster grid.

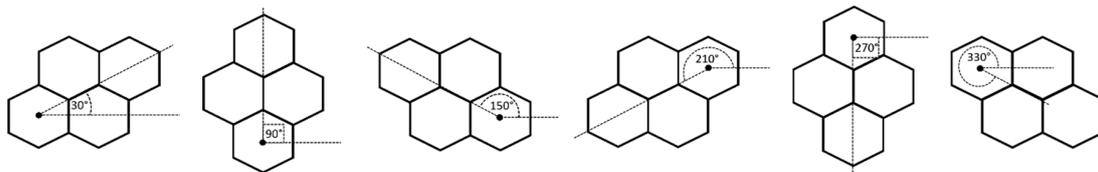
## 2.2. Cluster Plane Uniformity and Cluster Orientations

We define the cluster plane to be uniform if all clusters obtain the same orientation and size (area). The uniformity of the plane has the nice property that all the distance from the central cluster to all NN clusters is the same as given in Figure 4. With the hexagon of side length  $a$ , the center of the central cluster has coordinates  $S_0\left(\frac{3a\sqrt{3}}{4}, \frac{3a}{4}\right)$  and distance to all NN cluster centers is  $d = 2a\sqrt{3}$ .



**Figure 4.** The NN cluster distance and coordinates in the uniform 4-hexagonal cluster plane.

Clusters can have various orientations in the cluster plane, and we examine the possibilities with the rotation angles of step  $60^\circ$  ( $30^\circ, 90^\circ, 150^\circ, 210^\circ, 270^\circ, 330^\circ$ ). The example in Figure 4 represents the basic  $30^\circ$  angle of the 4-hexagonal cluster, and the remaining orientations are given in Figure 5.



**Figure 5.** Forming 4-hexagonal clusters and their orientations.

## 2.3. Research Problems

A problem of forming 4-hexagonal clusters is expressed by the following problem formulations:

- **Problem 1.** How to cluster the tessellated plane of hexagonal cells, such that the cluster plane remains uniform?
  - Is it possible to enclose all clusters inside a regular hexagonal container by using the existing models from the literature?
  - How to calculate the total number of inner clusters inside the container and the total number of shared clusters at the container border?
- **Problem 2.** How to derive a model from obtaining a new structure for hexagonal clustering, with clusters entirely embedded inside the container while keeping the cluster plane uniform?
- **Problem 3.** What is the efficiency of the proposed geometrical structure compared to the existing models in terms of the total number of packed clusters inside the container?

## 2.4. Research Context

In order to solve the above-formulated research problems, we concentrate on “bee” tetrahex clusters, having the minimal perimeter and that are rhomboid in shape (Figure 2b). It is assumed in our analysis that clustering starts with the central cluster contained in  $(0, 0)$ , i.e., that the clustering center overlaps with the center of the container (CA architecture), or with the cell that is closest to the container center (VA architecture). We consider the cluster orientations in the packing procedure,

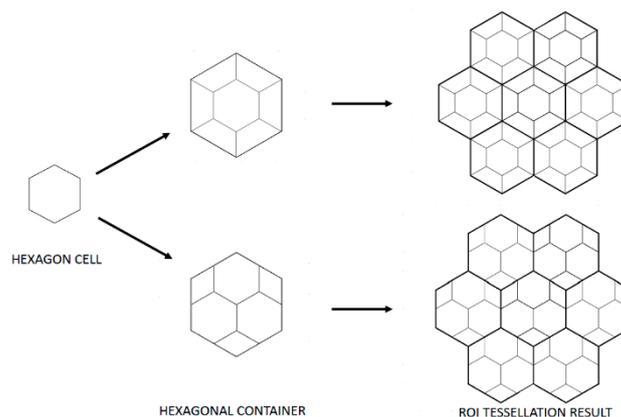
as shown in Figure 5. Our main intention with cluster packing is to keep the cluster plane uniform so that all clusters are the same in area and orientation, and the distance from the central cluster to its NNs is constant.

We apply clustering with existing models from the literature, CA, and VA structures, and we propose a VA model that enables all clusters to be packed in the container, with no partial clusters shared at the border. We calculate the number of different container types provided by each of the architectures when containers with the packed hexagonal clusters are tessellated, where our aim is to have a single container type. Also, our intention is to derive mathematical expressions from calculating the number of full (inner packed) and shared (border) clusters so that we can evaluate that indeed the proposed model is more efficient than the existing ones.

### 3. Clustering with the Existing Models

The two default variants of multi-resolution hexagonal grids that are often used in the literature are CA model, where small hexagonal cells share their midpoints with the container, and VA model, where the small-cell vertex overlaps with the container centroid [22,28,30].

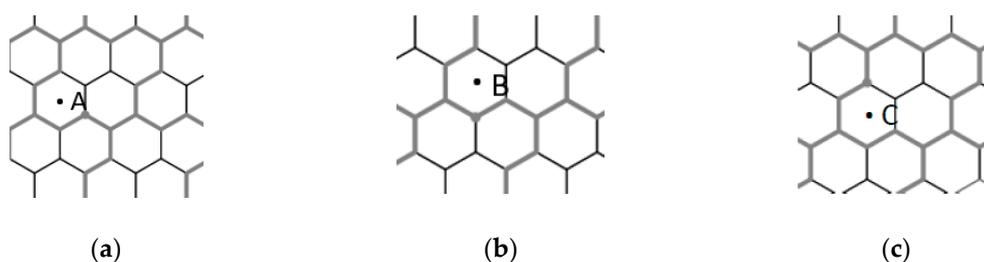
We apply a hexagonal tessellation approach to solve a packing problem of hexagonal clusters inside a regular hexagonal container. The basis is a regular hexagonal grid whose points are used as centers of the hexagonal cells. This is needed because, for cells to be clustered together, the container must consist of hexagonal cells packed inside (Figure 6). Next, the hexagonal ROIs must be tessellated in the higher-resolution grid, to examine the cluster plane uniformity when hexagonal clusters are formed.



**Figure 6.** Multi-resolution hexagonal grids. CA architecture (**up**) and VA architecture (**down**) (Adjusted from [28,30]).

We must choose the center of clustering accordingly. In the case of CA architecture, the center of clustering is straightforward; it is the center of the container (the center of a central hexagon). In VA architecture, the three central hexagons meet in the vertex that is a container center, and one of them needs to be chosen as a clustering center. There are three possibilities (Figure 7): cluster center-left (position A), cluster center-right (position B), and cluster center-down (position C). These three cases can be narrowed to only two since the selection A and B would give the same structure due to the symmetry of the container.

Considering the cluster rotations, in CA architecture, there are three pairs of clusters with different orientations having the same clustering structure. Those cluster orientation pairs are  $30^\circ$  and  $210^\circ$ ,  $90^\circ$  and  $270^\circ$ ,  $150^\circ$  and  $330^\circ$ . It is straightforward that the second clustering architecture in each pair is obtained by rotation of the first one for  $180^\circ$ . On the other hand, in the VA model, depending on the cluster center position, two different cases can be distinguished. For the cluster center moved down, the orientations  $30^\circ$ ,  $150^\circ$ ,  $210^\circ$ , and  $330^\circ$  all have the same clustering structure, while for cluster center moved left, the structure for orientations  $30^\circ$ ,  $90^\circ$ ,  $210^\circ$  and  $270^\circ$  is constant.



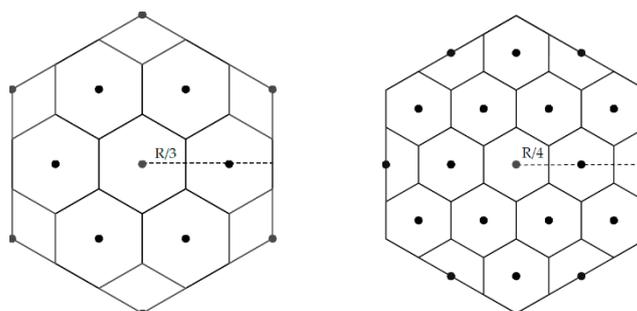
**Figure 7.** The options for the clustering center in the VA architecture. (a) Clustering center in cell A; (b) Clustering center in cell B; (c) Clustering center in cell C.

### 3.1. Centroid-Aligned Architecture

The formal description of the architectures for subdividing a container into a set of smaller hexagons is adopted from [12]. Authors show that the following proposition is valid:

**Proposition 1.** For each  $R/n$  subdivision in the centroid-aligned container, the number of equal hexagonal cells packed inside is  $n^2$ , where  $n$  is the subdivision level (Figure 8).

**Proof.** Refer to [12].  $\square$



**Figure 8.**  $R/3$  and  $R/4$  subdivisions for the CA architecture (Adjusted from [12]).

We adopt the described CA architecture and form 4-hexagonal clusters inside with using the existing model. We divide the model into two classes, depending on whether the subdivision level  $n$  is odd or even. We refer to those classes  $CA_{odd}$  and  $CA_{even}$ , respectively. Considering the requirement of the cluster plane uniformity, we examine all the cluster rotations from Figure 5.

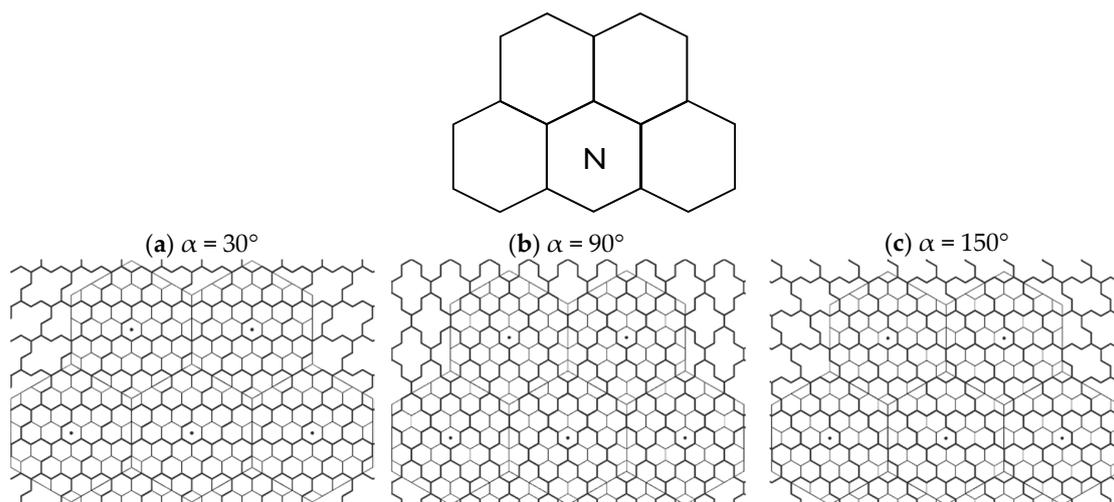
#### 3.1.1. Packing 4-Hexagonal Clusters in the Even-Sized CA Container

Let the architecture be  $CA_{even}$ , and the container size  $n = 2k$ ,  $k \in \mathbb{N}$ . Let us consider forming 4-hexagonal clusters in the uniform clustering approach and packing the clusters in the regular hexagonal container of even size  $CA_{even}$ .

**Proposition 2.** If  $CA_{even}$  containers are arranged in a tessellated manner, for default orientation ( $\alpha = 30^\circ$ ) they all have a single clustering structure.

**Proof.** The cluster contains two pairs of hexagons that are arranged in two adjacent rows like it is shown in Figure 9a. When the container size is even number (measured as the number of small hexagons at the double apothem), an integer number of hexagon pairs are positioned between two opposite hexagon sides, which results in the same clustering structure.  $\square$

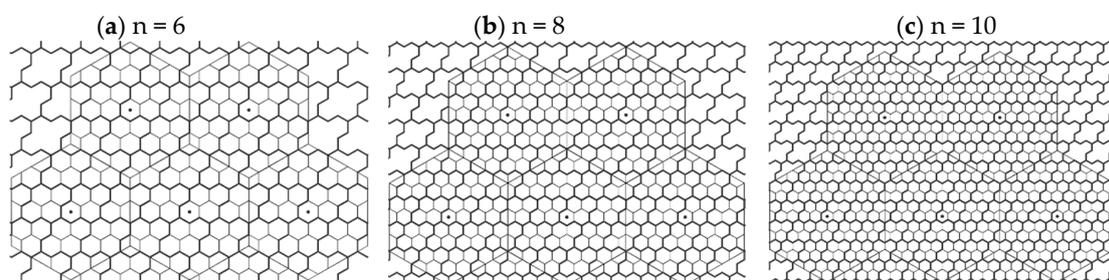
**Proposition 3.** No matter of the cluster rotation, there is a single structure of the tessellated  $CA_{even}$  containers with the inner packed cluster items, i.e., in Proposition 2, the rotation invariance is valid.



**Figure 9.** 4-hexagonal CA clustering result for even container  $n = 6$  and cluster orientation  $30^\circ, 90^\circ, 150^\circ$ . A single container type  $N$  is shown.

**Proof.** We denote a single container type as container  $N$  (Figure 9). According to Proposition 2,  $CA_{\text{even}}$  containers with  $30^\circ$  orientation will have a single clustering structure. Cluster rotation is available for rotation angles of step  $60^\circ$ . Since a container shape is a regular hexagon, it has rotational symmetry (of order six) with rotation angle  $60^\circ$ . Hence, a container structure remains constant independent of the cluster orientation.  $\square$

Clustering results from Figure 10 are summarized in Table 1, providing the number of the packed inner full clusters inside the container and shared clusters at the border, when the default  $30^\circ$  cluster orientation is considered.



**Figure 10.** 4-hexagonal CA clustering result for even container size and cluster orientation  $30^\circ$ .

**Table 1.** Even container size in the 4-hexagonal CA clustering model.

Container Size ( $n$ )	#Full (Inner) Clusters	#Border (Shared) Clusters
6	5	10
8	10	13
10	17	17
12	27	20
14	38	23
16	51	27

Let us derive a single algebraic expression that covers all even container sizes for calculating the number of inner packed cluster items:

$$N_{full} = (n/2 - 1)^2 + \lfloor n/6 \rfloor \tag{1}$$

Authors in [12] prove that as the levels of subdivision  $n$  increase, the sub-cells inside (in a one-sixth of a hexagonal container) increase by one for every three levels. This means that there are several sub-clustering architectures inside the larger class  $CA_{\text{even}}$ . Motivated by that finding, and the clustering visualizations such as the ones in Figure 10, we conclude that the clustering structure depends on whether the container size  $n$  is a multiple of three, i.e., we can divide the  $CA_{\text{even}}$  class in three sub-classes  $CA_{\text{even}}(r)$  based on the remainder of dividing by three (marked with  $r$ ), where  $n = 3k + r$ ,  $k \in \mathbb{N}$ ,  $r \in \{0, 1, 2\}$ .

**Proposition 4.** *There will always be three different clustering structures depending if the even CA container size  $n$  is a multiple of 3.*

**Proof.** Having a container of size  $n$ , the distance from the center of the container to any vertex is equal to  $n * d$ , where  $d$  is the diameter of a small hexagon. Depending on the  $n$  value, the path from container center to container vertex is the sequence of vertices A (beginning of the hexagon), B (center of the hexagon), C (end of the hexagon), A, B ... like is shown in Figure 11. □

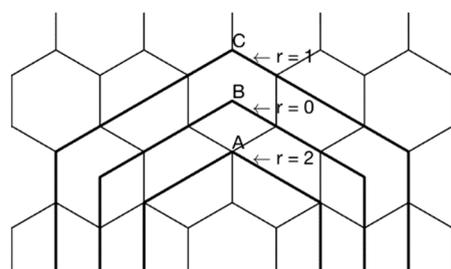


Figure 11. CA container example and the corresponding path.

Positions A, B, C alternate sequentially with the increase of container size. Therefore, it is possible to distinguish three different container vertex positions that result in three different clustering architectures depending on the remainder  $r$  of dividing  $n$  by three. Note that this is valid for any cluster rotation. The recursive formula is valid, covering all CA even containers independent on their type:

$$N_{full_{i+1}} = \begin{cases} N_{full_i} + n_i, & \text{for } r = 0 \\ N_{full_i} + n_i - 1, & \text{else} \end{cases} \quad (2)$$

In Equation (2), the index  $i$  is an iteration number in the recursion  $i \in \mathbb{N}_0$ , with the initial conditions  $n_0 = 4$  and  $N_{full_0} = 1$ . Additionally, we derive separate algebraic expressions for calculating the number of inner packed and shared clusters for each architecture type  $CA_{\text{even}}(r)$ .

The number of full clusters  $N_{full}$  is:

$$N_{full} = \begin{cases} (3n^2 - 10n + 12)/12, & \text{if } r = 0 \\ (3n^2 - 10n + 4)/12, & \text{if } r = 1 \\ (3n^2 - 10n + 8)/12, & \text{if } r = 2 \end{cases} \quad (3)$$

The number of shared clusters  $N_{shared}$  is calculated with the following formula:

$$N_{shared} = \begin{cases} (5n)/3, & \text{if } r = 0 \\ (5n + 1)/3, & \text{if } r = 1 \\ (5n - 1)/3, & \text{if } r = 2 \end{cases} \quad (4)$$

**Cluster Rotation Invariance:** As it can be seen in Figure 9, the number of inner clusters  $N_{full}$  and  $N_{shared}$  is constant for all cluster rotations (refer to Proposition 3).

### 3.1.2. Packing 4-Hexagonal Clusters in the Odd-Sized CA Container

Let the architecture be  $CA_{\text{odd}}$ , and the container size  $n = 2k + 1$ ,  $k \in N$ . Let us consider forming 4-hexagonal clusters in the uniform clustering approach and packing the clusters in the regular hexagonal container of odd size  $CA_{\text{odd}}$ .

**Proposition 5.** *If  $CA_{\text{odd}}$  containers are arranged in a tessellated manner, for default orientation ( $\alpha = 30^\circ$ ) there are two types of clustering container structures.*

**Proof.** The cluster contains two pairs of hexagons that are arranged in two adjacent rows. When a container size is an odd number, the width of two adjacent clusters is even number (Figure 12), which means that the third cluster will have the same structure as the first one. □

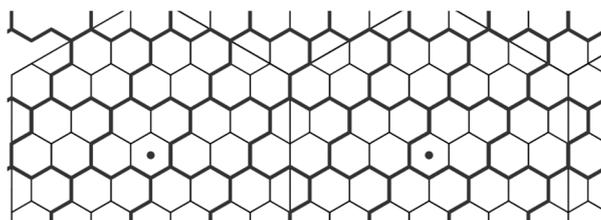


Figure 12. The two adjacent clusters in  $CA_{\text{odd}}$  architecture.

Therefore, clusters in the current cluster row have two different structures that alternate (M and N in Figure 13). Clusters in adjacent rows have the same structure but rotated for  $180^\circ$ .

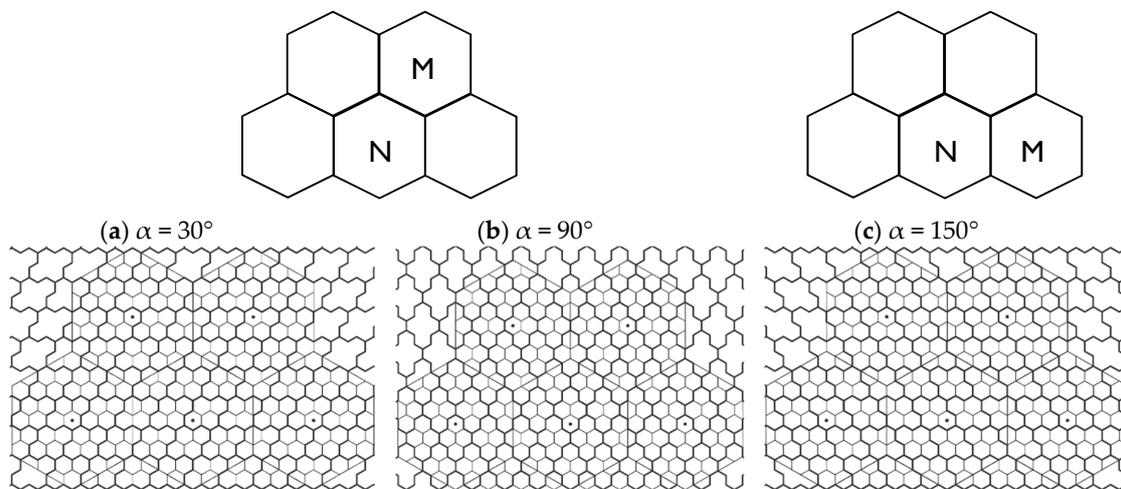


Figure 13. 4-hexagonal CA clustering result for odd container  $n = 7$  and cluster orientation  $30^\circ$ ,  $90^\circ$ ,  $150^\circ$ . Two container types N and M are shown.

**Proposition 6.** *No matter which cluster rotation is considered, there will be two clustering container types when  $CA_{\text{odd}}$  containers with the inner packed cluster items are tessellated, i.e., in Proposition 5, the rotational invariance is valid.*

**Proof.** Like in Proposition 3, cluster rotation with angles of step  $60^\circ$  packed in the container with rotational symmetry (of order six) with an angle of rotation  $60^\circ$  results in the same clustering structure when rotated again for the step of  $60^\circ$  angle. Consequently, there will always be two types of clustering containers invariant of the cluster orientation. □

Clustering results from Figure 14 are summarized in Table 2, providing the number of packed inner full clusters inside the container and shared clusters at the border. It is done for both container

types  $N$  and  $M$ , where the calculated numbers are denoted as  $N_{full}$  and  $M_{full}$ , and cluster orientation  $30^\circ$  are considered.

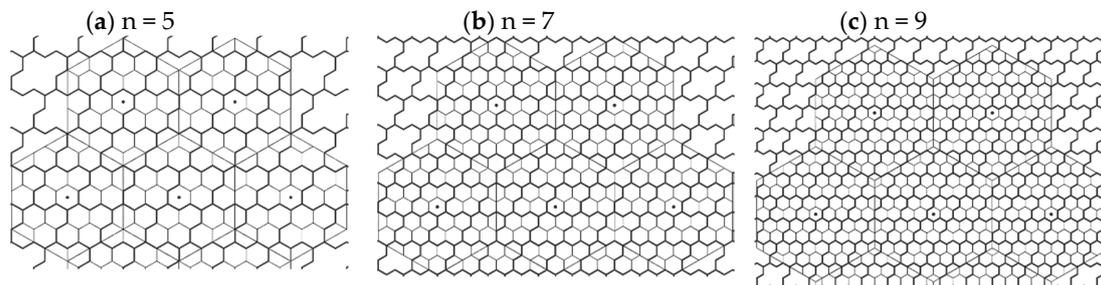


Figure 14. 4-hexagonal CA clustering result for odd container size and cluster orientation  $30^\circ$ .

Table 2. Odd container size in the 4-hexagonal CA clustering model.

Container Size ( $n$ )	#Full (Inner) Clusters		#Border (Shared) Clusters	
	$N_{full}$	$M_{full}$	$N_{shared}$	$M_{shared}$
5	3	2	9	9
7	6	7	13	11
9	12	14	18	15
11	21	21	21	19
13	30	32	25	21
15	42	45	30	25

Again, by using the same approach, we divide the  $CA_{odd}$  class in three sub-classes  $CA_{odd}(r)$  based on the remainder of dividing by three, where  $n = 3k + r, k \in N, r \in \{0, 1, 2\}$ . Also, there will always be three different clustering structures depending if the odd CA container size  $n$  is a multiple of three (refer to Proposition 4). The recursive formula is valid for  $N_{full}$ :

$$N_{full_{i+1}} = \begin{cases} N_{full_i} + n_i - 1, & \text{for } r = 0 \\ N_{full_i} + n_{i-1}, & \text{for } r = 1 \\ N_{full_i} + n_i, & \text{for } r = 2 \end{cases} \tag{5}$$

In Equation (5), the index  $i$  is an iteration number in the recursion  $i \in N_0$ , with the initial conditions  $n_0 = 3$  and  $N_{full_0} = 0$ .

The number of clusters inside the central container  $N$  is:

$$N_{full} = \begin{cases} (n^2 - 4n + 7)/4, & \text{if } r = 2 \\ (n^2 - 4n + 3)/4, & \text{else} \end{cases} \tag{6}$$

and formula for  $M_{full}$  is obtained:

$$M_{full} = \begin{cases} (3n^2 - 10n + 15)/12, & \text{if } r = 0 \\ (3n^2 - 10n + 7)/12, & \text{if } r = 1 \\ (3n^2 - 10n - 1)/12, & \text{if } r = 2 \end{cases} \tag{7}$$

Next, the number of shared clusters  $N_{shared}$  and  $M_{shared}$  is derived, respectively:

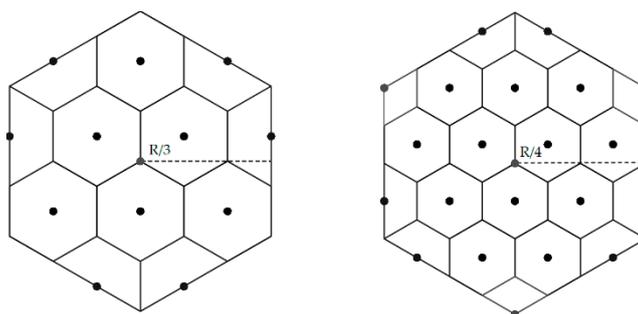
$$N_{shared} = \begin{cases} 2n, & \text{if } r = 0 \\ 2n - 1, & \text{else} \end{cases} \tag{8}$$

$$M_{shared} = \begin{cases} (5n)/3, & \text{if } r = 0 \\ (5n - 2)/3, & \text{if } r = 1 \\ (5n + 2)/3, & \text{if } r = 2 \end{cases} \quad (9)$$

*Cluster Rotation Invariance:* As can be seen in Figure 13, the number of inner clusters  $N_{full}$ ,  $M_{full}$ , and  $N_{shared}$ ,  $M_{shared}$  is constant for all cluster rotations (refer to Propositions 5 and 6).

### 3.2. Vertex-Aligned Architecture

We use VA architecture derived from the CA model, where the container is moved such that its center does not overlap with the center but with the vertex of the central hexagonal cell (Figure 15). By applying the same notation of the  $R/n$  subdivision adopted from [12], we conclude that the same Proposition 1 about the total number of inner-packed equal hexagonal cells' number is valid.



**Figure 15.** R/3 and R/4 subdivisions for VA architecture.

We adopt the described VA architecture and form 4-hexagonal clusters inside with using this existing model. Again, we divide the model into two classes, depending on whether the subdivision level  $n$  is odd or even. We refer to those classes  $VA_{odd}$  and  $VA_{even}$ , respectively. Considering the requirement of the cluster plane uniformity, we examine all the cluster rotations from Figure 5.

#### 3.2.1. Packing 4-Hexagonal Clusters in the Even-Sized VA Container

Let the architecture be  $VA_{even}$ , and the container size  $n = 2k$ ,  $k \in N$ . Let us consider forming 4-hexagonal clusters in the uniform clustering approach and packing the clusters in the regular hexagonal container of even size  $VA_{even}$ .

**Proposition 7.** *If  $VA_{even}$  containers are arranged in a tessellated manner, for default orientation ( $\alpha = 30^\circ$ ) they all have a single clustering structure.*

**Proof.** Refer to Proposition 2.  $\square$

**Proposition 8.** *No matter which cluster rotation is considered, there will be a single structure of the tessellated  $VA_{even}$  containers with the inner packed clusters, i.e., in Proposition 7, rotation invariance is valid.*

**Proof.** Refer to Proposition 2 and 3.  $\square$

**Proposition 9.** *No matter which center of clustering is considered (center moved down, or center moved left/right), there will be a single structure of the tessellated  $VA_{even}$  containers with the inner packed cluster items.*

**Proof.** The tessellation is independent of the clustering center position. Therefore, the structure of the containers remains the same.  $\square$

VA architecture with different cluster center position is shown on Figures 16 and 17. Quantitative properties of the clustering results from Figures 18 and 19 are summarized in Table 3, providing the

number of the packed inner full clusters inside the container and shared clusters at the border, when the default 30° cluster orientation is considered.

**Table 3.** Even container size in the 4-hexagonal VA clustering model.

Container Size (n)	#Full (Inner) Clusters	#Border (Shared) Clusters
6	4	11
8	9	15
10	16	20
12	25	23
14	36	27
16	49	32

We derive a single algebraic expression for all even VA container sizes for calculating the number of packed cluster items:

$$N_{full} = (n/2 - 1)^2 \tag{10}$$

Despite a single Equation (10) is derived, the clustering structure still depends on whether the container size  $n$  is a multiple of three. It is because the same Proposition 1 is valid in VA as in a centralized case. Therefore, we can divide the  $VA_{even}$  class into three sub-classes  $VA_{even}(r)$  based on the remainder of dividing by three (marked with  $r$ ), where  $n = 3k + r, k \in N, r \in \{0, 1, 2\}$ . Also, there will always be three different clustering structures depending if the even VA container size  $n$  is a multiple of 3 (refer to Proposition 4).

We derive formulas for calculating the number of shared clusters for each architecture type  $VA_{even}(r)$ . The number of shared hexagonal clusters on the container border  $N_{shared}$  is:

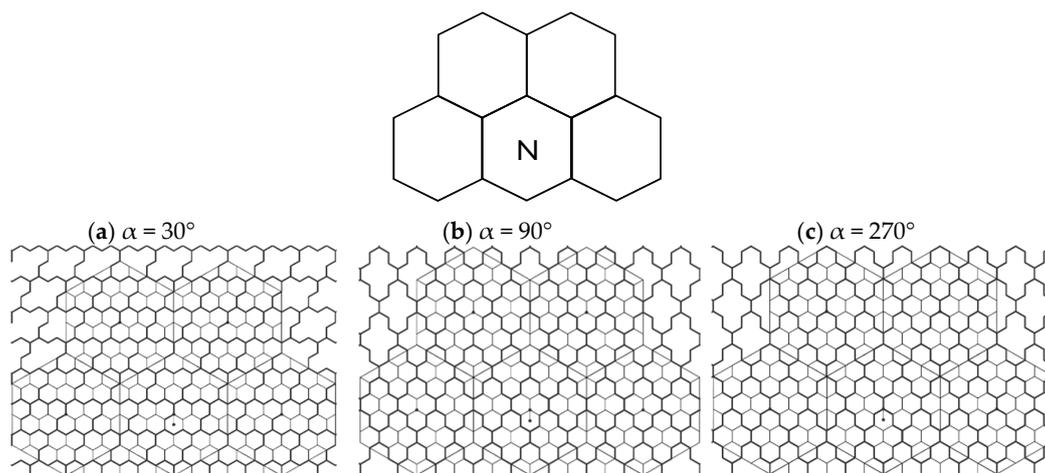
$$N_{shared} = \begin{cases} 2n, & \text{if } r = 1 \\ 2n - 1, & \text{else} \end{cases} \tag{11}$$

*Cluster Rotation Invariance:* As it can be seen in Figure 16 (for cluster center down) and Figure 17 (for cluster center left), the number of inner clusters  $N_{full}$  and  $N_{shared}$  is not the same for all cluster rotations. The rotation invariance is summarized in Table 4. The existing formulas are applied in almost all cases, while for cluster rotation angle  $\alpha = 270^\circ$  and the architecture moved down, the following formula is used to calculate the number of full clusters:

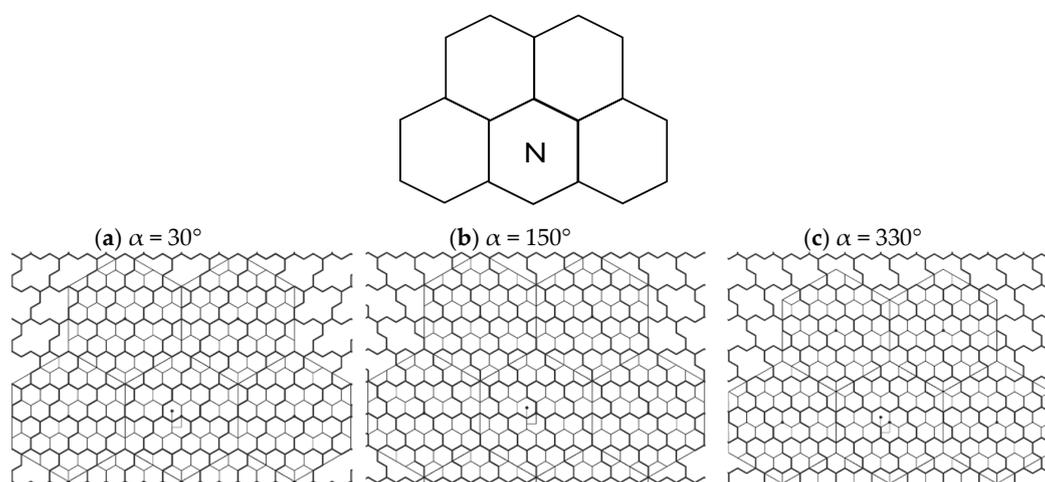
$$N_{full} = \begin{cases} (3n^2 - 10n)/12, & \text{if } r = 0 \\ (3n^2 - 10n + 16)/12, & \text{if } r = 1 \\ (3n^2 - 10n + 8)/12, & \text{if } r = 2 \end{cases} \tag{12}$$

**Table 4.** The cluster rotation invariance for  $VA_{even}$  architectures.

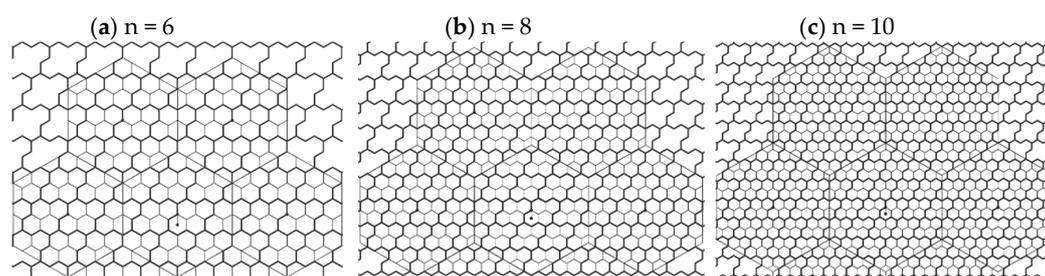
	$VA_{even}$ Moved Down	$VA_{even}$ Moved Left
	$N_{full}, N_{shared}$	$N_{full}, N_{shared}$
$\alpha = 30^\circ$	Formula (10), (11)	Formula (10), (11)
$\alpha = 90^\circ$	Formula (1), (4)	Formula (10), (11)
$\alpha = 270^\circ$	Formula (12), (9)	Formula (10), (11)
$\alpha = 150^\circ$	Formula (10), (11)	Formula (12), (9)
$\alpha = 330^\circ$	Formula (10), (11)	Formula (1), (4)



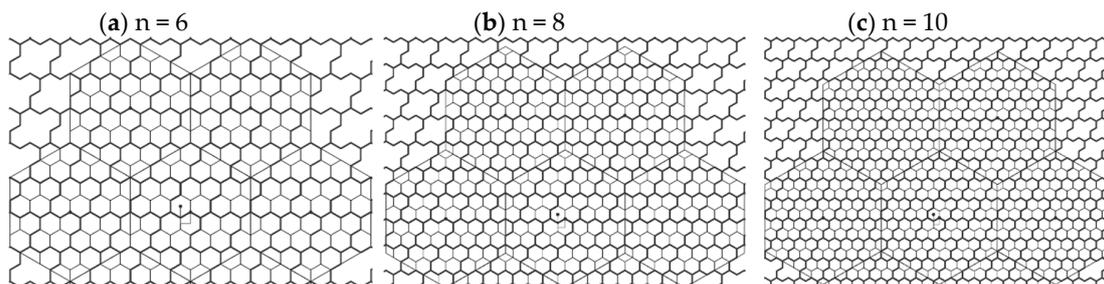
**Figure 16.** 4-hexagonal VA clustering result for even container  $n = 6$  and cluster orientation  $30^\circ$ ,  $90^\circ$ ,  $270^\circ$ . A single container type N is shown (central cluster position down).



**Figure 17.** 4-hexagonal VA clustering result for even container  $n = 6$  and cluster orientation  $30^\circ$ ,  $150^\circ$ ,  $330^\circ$ . A single container type N is shown (central cluster position left).



**Figure 18.** 4-hexagonal VA clustering result for even container size and cluster orientation  $30^\circ$  (central cluster position down).



**Figure 19.** 4-hexagonal VA clustering result for even container size and cluster orientation  $30^\circ$  (central cluster position left).

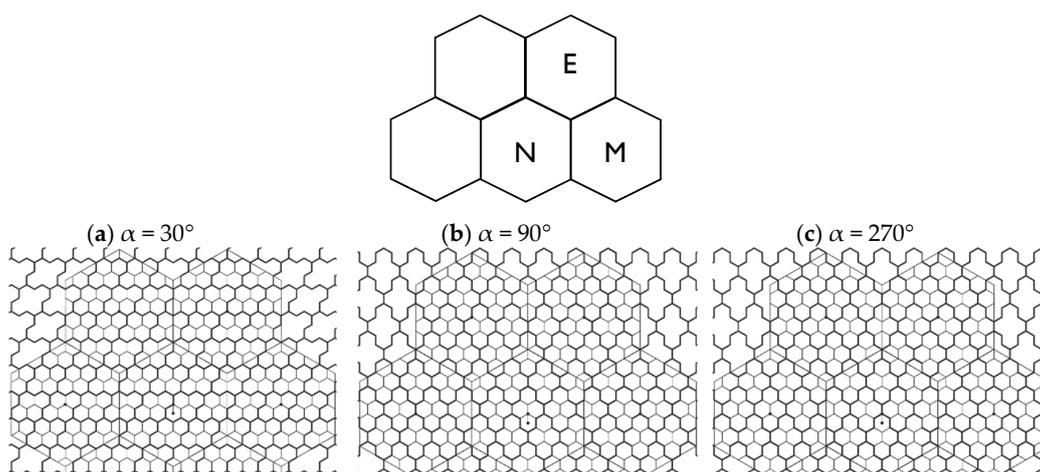
### 3.2.2. Packing 4-Hexagonal Clusters in the Odd-Sized VA Container

Let the architecture be  $VA_{\text{odd}}$ , and the container size  $n = 2k + 1$ ,  $k \in N$ . Let us consider forming 4-hexagonal clusters in the uniform clustering approach and packing the clusters in the regular hexagonal container of odd size  $VA_{\text{odd}}$ .

**Proposition 10.** *There are three types of clustering container structures when  $VA_{\text{odd}}$  containers with packed cluster items are arranged in a tessellated manner for default orientation ( $\alpha = 30^\circ$ ).*

**Proof.** Like Proposition 5, if container size is an odd number, the width of two adjacent clusters is even number meaning that the third cluster will have the same structure as the first one. Therefore, clusters in the current cluster row have two different structures (N and M). Clusters in the adjacent row also have two different structures that alternate, but one of them is reflectively symmetrical with one of structures N or M. Hence, altogether, there are three types of clustering container structures.  $\square$

**Proposition 11.** *No matter which cluster rotation is considered, there will be three clustering container types when  $VA_{\text{odd}}$  containers with the inner packed cluster items are tessellated, i.e., the Proposition 10 is rotation invariant (Figure 20).*

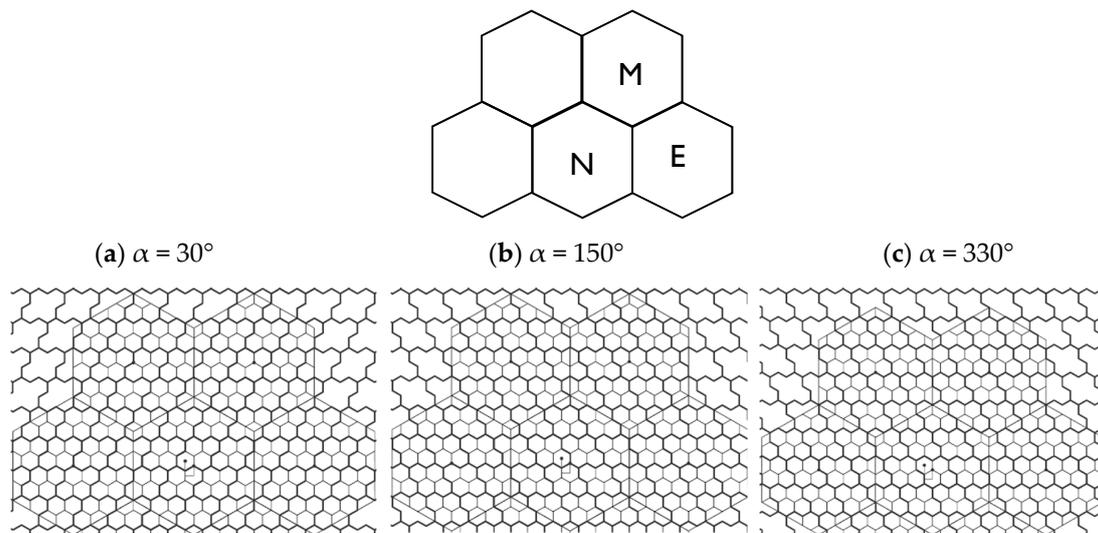


**Figure 20.** 4-hexagonal VA clustering result for odd container  $n = 7$  and cluster orientation  $30^\circ, 90^\circ, 270^\circ$ . Three container types N, M, and E are shown (central cluster position down).

**Proof.** Refer to Proposition 6.  $\square$

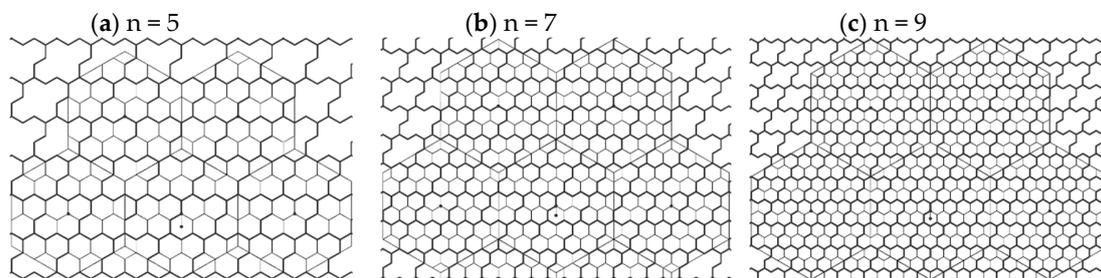
**Proposition 12.** *No matter which center of clustering is considered (center moved down, or center moved left/right), there will be three different structures of the tessellated  $VA_{\text{even}}$  containers with the inner packed cluster items (Figure 21).*

**Proof.** Refer to Proposition 9.  $\square$

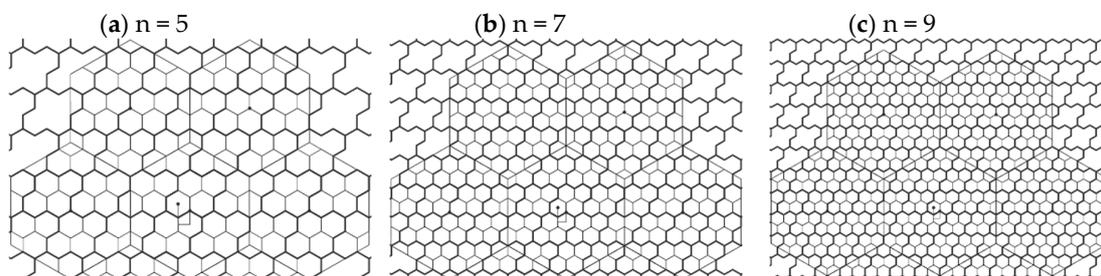


**Figure 21.** 4-hexagonal VA clustering result for odd container  $n = 7$  and cluster orientation  $30^\circ$ ,  $150^\circ$ ,  $330^\circ$ . Three container types N, M, and E are shown (central cluster position left).

Clustering results from Figures 22 and 23 are summarized in Table 5, providing the number of the packed inner full clusters inside the container and shared clusters at the border. It is shown for all three container types N, M, and E, denoted as  $N_{full}$ ,  $M_{full}$ , and  $E_{full}$ , where the default  $\alpha = 30^\circ$  cluster orientation is considered. Also, there will always be three different clustering structures depending if the odd VA container size  $n$  is a multiple of three (refer to Proposition 4).



**Figure 22.** 4-hexagonal VA clustering result for odd container size and cluster orientation  $30^\circ$  (central cluster position down).



**Figure 23.** 4-hexagonal VA clustering result for odd container size and cluster orientation  $30^\circ$  (central cluster position left).

**Table 5.** Odd container size in the 4-hexagonal VA clustering model.

Container Size (n)	#Full (Inner) Clusters			#Border (Shared) Clusters		
	$N_{full}$	$M_{full}$	$E_{full}$	$N_{shared}$	$M_{shared}$	$E_{shared}$
5	3	1	3	8	10	9
7	7	7	6	12	12	15
9	13	13	12	15	18	16
11	22	19	21	18	22	21
13	32	31	30	22	24	27
15	44	43	42	25	30	28

Again, by using the same approach, we divide the  $VA_{\text{odd}}$  class in three sub-classes  $VA_{\text{odd}}(r)$  based on the remainder of dividing by 3, where  $n = 3k + r$ ,  $k \in N$ ,  $r \in \{0, 1, 2\}$ . The recursive formula is valid for  $N_{full}$ :

$$N_{full_{i+1}} = \begin{cases} N_{full_i} + n_i, & \text{for } r = 2 \\ N_{full_i} + n_i - 1, & \text{else} \end{cases} \quad (13)$$

In Equation (13), the index  $i$  is an iteration number in the recursion,  $i \in N_0$ , with the initial conditions  $n_0 = 3$  and  $N_{full_0} = 0$ .

The number of full hexagonal clusters  $N_{full}$  packed inside the container is:

$$N_{full} = \begin{cases} (3n^2 - 10n + 3)/12, & \text{if } r = 0 \\ (3n^2 - 10n + 7)/12, & \text{if } r = 1 \\ (3n^2 - 10n + 11)/12, & \text{if } r = 2 \end{cases} \quad (14)$$

$E_{full}$  is calculated in formula (6), while formulas for calculating  $M_{full}$  are:

$$M_{full} = \begin{cases} (n^2 - 4n - 1)/4, & \text{if } r = 2 \\ (n^2 - 4n + 7)/4, & \text{else} \end{cases} \quad (15)$$

Next, the number of shared clusters for each container type, i.e.,  $N_{shared}$ ,  $M_{shared}$  and  $E_{shared}$  is derived in continuation:

$$N_{shared} = \begin{cases} (5n)/3, & \text{if } r = 0 \\ (5n + 1)/3, & \text{if } r = 1 \\ (5n - 1)/3, & \text{if } r = 2 \end{cases} \quad (16)$$

$$M_{shared} = \begin{cases} 2n - 2, & \text{if } r = 1 \\ 2n, & \text{else} \end{cases} \quad (17)$$

$$E_{shared} = \begin{cases} 2n - 2, & \text{if } r = 0 \\ 2n + 1, & \text{if } r = 1 \\ 2n - 1, & \text{if } r = 2 \end{cases} \quad (18)$$

*Cluster Rotation Invariance:* As it can be seen in Figure 20 (for cluster center down) and Figure 21 (for cluster center left), the number of inner clusters  $N_{full}$ ,  $M_{full}$ ,  $E_{full}$  and  $N_{shared}$ ,  $M_{shared}$ ,  $E_{shared}$  is not the same for all cluster rotations. The rotation invariance is summarized in Table 6, where we can see that formulas are permuted depending on the chosen  $N$ ,  $M$  and  $E$  type.

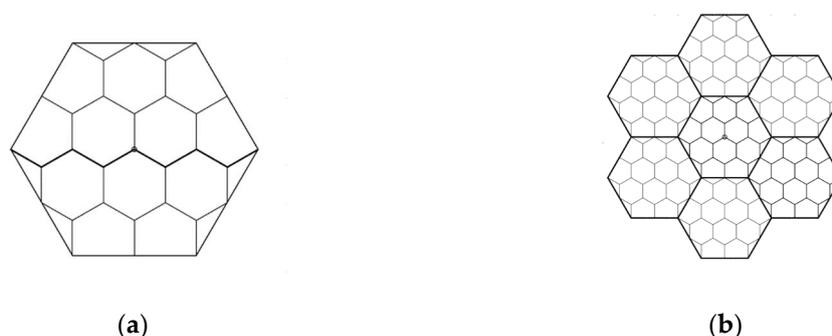
**Table 6.** The cluster rotation invariance for  $VA_{odd}$  architectures.

	$VA_{odd}$ Moved Down	$VA_{odd}$ Moved Left
	$N_{full}, M_{full}, E_{full}, N_{shared}, M_{shared}, E_{shared}$	$N_{full}, M_{full}, E_{full}, N_{shared}, M_{shared}, E_{shared}$
$\alpha = 30^\circ$	Formula (14), (15), (6), (16), (17), (18)	Formula (6), (15), (14), (18), (17), (16)
$\alpha = 90^\circ$	Formula (15), (6), (14), (17), (18), (16)	Formula (6), (15), (14), (18), (17), (16)
$\alpha = 270^\circ$	Formula (6), (15), (14), (18), (17), (16)	Formula (6), (15), (14), (18), (17), (16)
$\alpha = 150^\circ$	Formula (14), (15), (6), (16), (17), (18)	Formula (6), (15), (14), (18), (17), (16)
$\alpha = 330^\circ$	Formula (14), (15), (6), (16), (17), (18)	Formula (15), (6), (14), (17), (18), (16)

#### 4. The Proposed Vertex-Aligned Architecture

##### 4.1. General Description

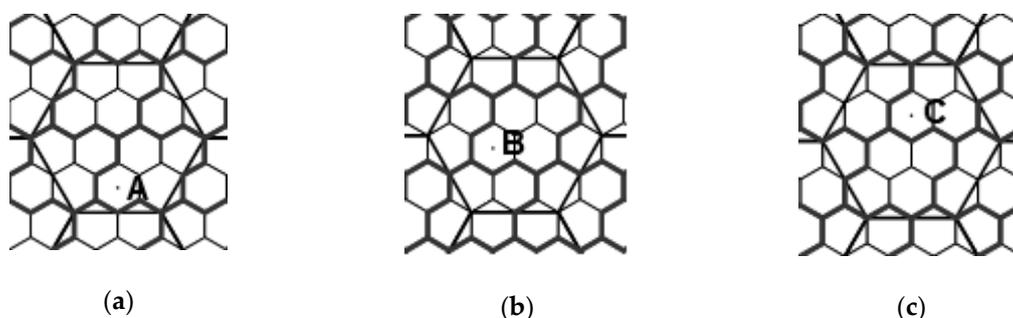
We propose a structure for more efficient packing of the 4-hexagonal clusters in the regular hexagonal container. It is adapted from [29], where its formal definition is described. Namely, the model can be referred to as  $H(D)$ , where  $D = 8k$ ,  $k \in N$  is a total number of inner small hexagonal edges, situated on the circumscribed circle diameter of the hexagonal container. Example for  $D = 8$  and the corresponding tessellation results are given in Figure 24.



**Figure 24.** The proposed container structures. (a) container model  $H(8)$  with marked edges; (b) the tessellation of containers (Adjusted from [29]).

##### 4.2. Uniform Clustering with the Proposed Model

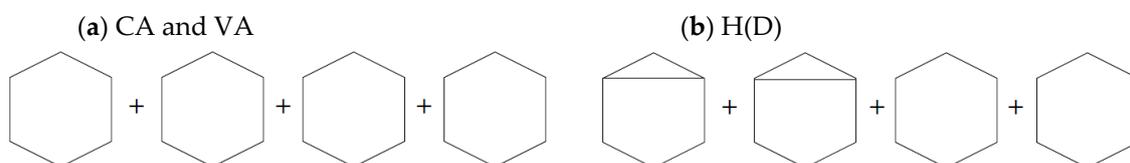
We propose a uniform clustering with the proposed model. Again, like in the case of CA and VA, we choose one of the hexagonal cells that is closest to the real container center to be the clustering center. As shown in Figure 25, we have three possibilities: central cluster position down, central cluster position left (symmetric to position right), and central cluster position up.



**Figure 25.** Center of clustering positions. (a) down; (b) left; (c) up.

It can be seen, for instance, in Figure 25a that there will be some compromises at the container border. Namely, instead of four full hexagons forming the 4-hexagonal cluster, some clusters at the

container border will be smaller in area, as they consist of two full hexagons and two truncated hexagon pieces (Figure 26). However, the difference in the cluster area is acceptable (9.1%). Also, there are voids on some container edges, given in the form of several cut-out triangles.



**Figure 26.** The comparison of cluster area for CA, VA, and the proposed model.

#### 4.2.1. The 4-Hexagonal Cluster Rotations

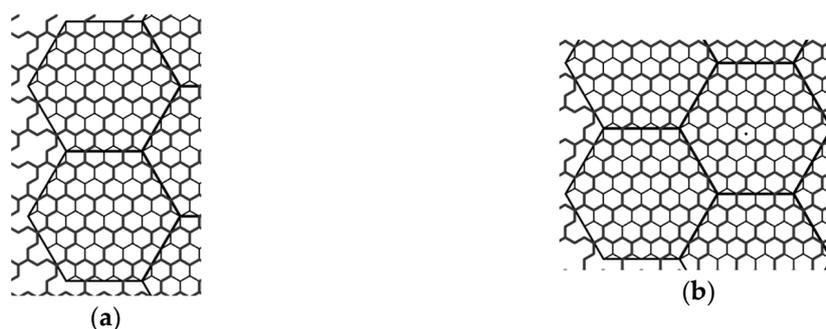
If we consider the cluster rotations for this type of architecture, depending of the cluster center position, different cases can be distinguished. For the cluster center position down, two groups of orientations having the same clustering structure are the first one  $30^\circ$ ,  $150^\circ$  and  $270^\circ$  and the other one  $90^\circ$ ,  $210^\circ$  and  $330^\circ$ . In the case of cluster center position left, there are three (groups of) orientations having the same structure:  $30^\circ$ , then  $90^\circ$ ,  $150^\circ$ ,  $270^\circ$ , and  $330^\circ$  are the same, and the third type is  $210^\circ$  orientation. Finally, for cluster center position up, there are three possible (groups of) orientations with constant structure:  $30^\circ$ , then  $90^\circ$ ,  $150^\circ$ ,  $210^\circ$ , and  $330^\circ$  having the same structure, and the last one type is  $270^\circ$  orientation.

#### 4.2.2. The Geometrical Properties and Derived Formulas

Let the architecture be  $H(D)$ , and the container size  $D = 8k$ ,  $k \in \mathbb{N}$ . Let us consider forming 4-hexagonal clusters in the uniform clustering approach and packing the clusters in the proposed regular hexagonal container.

**Proposition 13.** *If  $H(D)$  containers are arranged in a tessellated manner, for default orientation ( $\alpha = 30^\circ$ ), they all have a single clustering structure.*

**Proof.** Like in the case of even-sized CA container type, since  $D = 8k$ , the  $H(D)$  container contains even number of hexagons between two opposite hexagon sides. Therefore, all containers in the current container column have the same structure (Figure 27a). The containers in adjacent columns are the same as the previous one but translated, as shown in Figure 27b.  $\square$



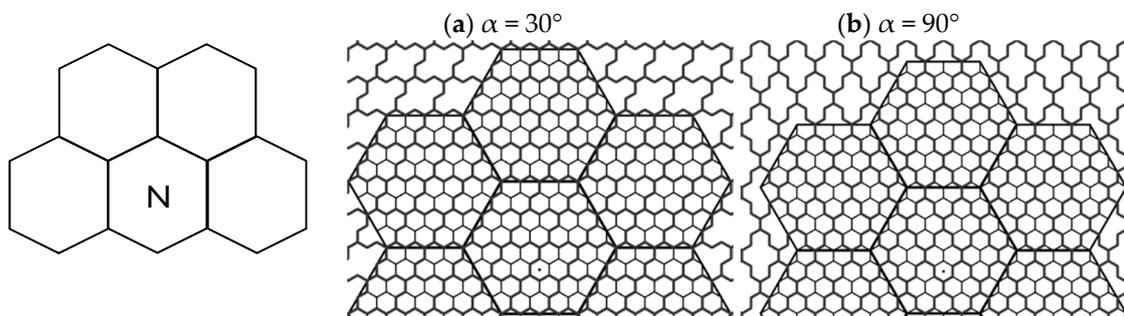
**Figure 27.** (a) Container column; (b) Containers in adjacent column.

**Proposition 14.** *No matter which cluster rotation is considered, there will be a single structure of the tessellated  $H(D)$  containers with the inner packed cluster items, i.e., in Proposition 13, rotation invariance is valid.*

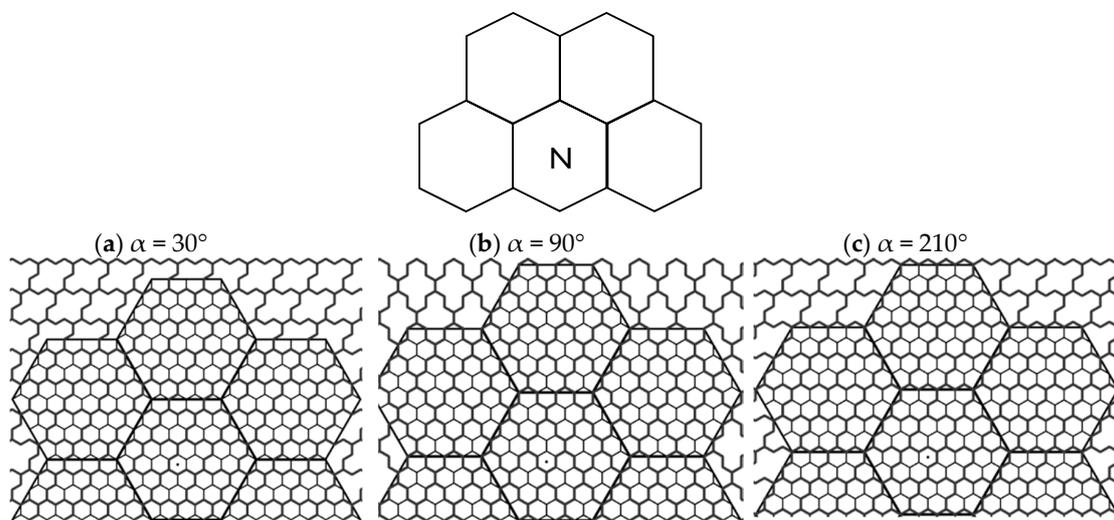
**Proof.** Refer to Proposition 3.  $\square$

**Proposition 15.** *No matter which center of clustering is considered (center moved down/up, or center moved left/right), there will be a single structure of the tessellated  $H(D)$  containers with the inner packed cluster items.*

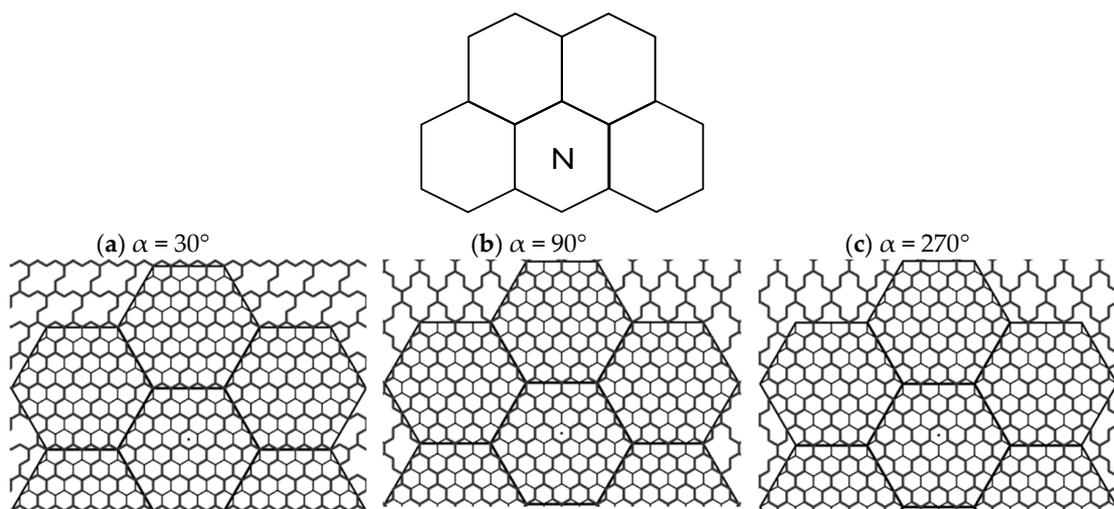
**Proof.** The tessellation is independent of the clustering center position. Therefore, the choice of clustering center doesn't have an influence on the containers structure (Figures 28–30).  $\square$



**Figure 28.** 4-hexagonal H(D) clustering result for container size  $D = 16$  and cluster orientation  $30^\circ$  and  $90^\circ$ . A single container type is shown (central cluster position down).

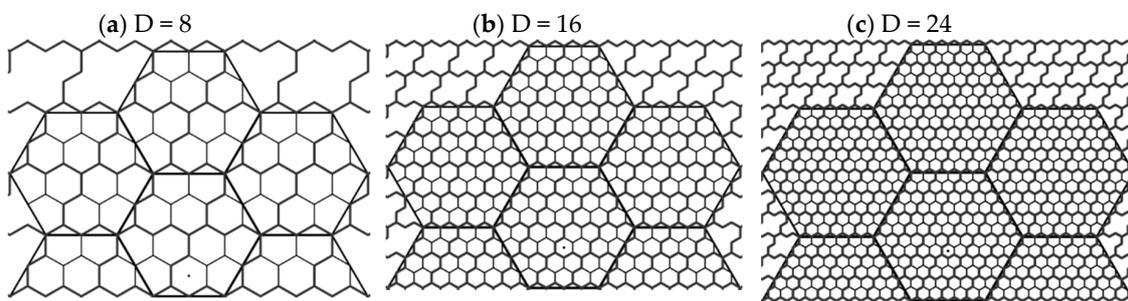


**Figure 29.** 4-hexagonal H(D) clustering result for container size  $D = 16$  and cluster orientation  $30^\circ$ ,  $90^\circ$ , and  $210^\circ$ . Single container type is shown (central cluster position left).



**Figure 30.** 4-hexagonal H(D) clustering result for container size  $D = 16$  and cluster orientation  $30^\circ$ ,  $90^\circ$ , and  $270^\circ$ . Single container type is shown (central cluster position up).

Clustering results from Figure 31 are summarized in Table 7, providing the number of inner full clusters packed inside the container and the number of shared clusters at the border, when the default 30° cluster orientation is considered.



**Figure 31.** 4-hexagonal  $H(D)$  clustering result for container size  $D = \{8, 16, 24\}$  and cluster orientation 30° (central cluster position down).

**Table 7.** The proposed 4-hexagonal  $H(D)$  clustering model.

Container Size (D)	#Full (Inner) Clusters	#Border (Shared) Clusters
8	2	2
16	10	4
24	24	6
32	44	8
40	70	10

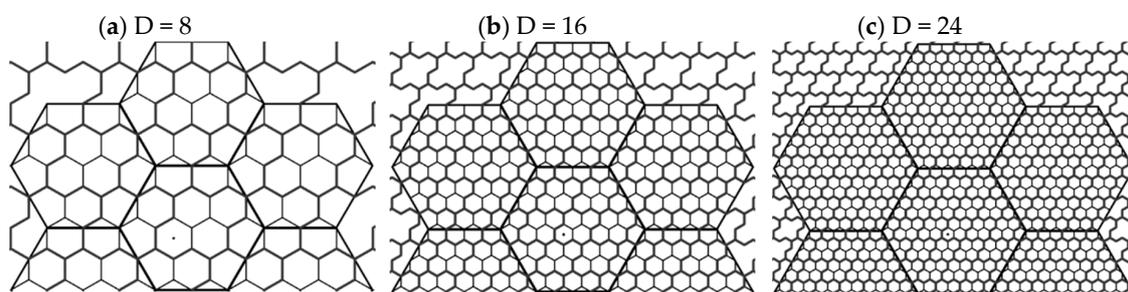
An algebraic expression for calculating the number of inner packed cluster items is as follows:

$$N_{full} = (3D^2 - 8D)/64 \tag{19}$$

Also, the number of shared clusters at the container border is calculated as:

$$N_{shared} = D/4 \tag{20}$$

Clustering results from Figure 32 are summarized in Table 8, providing the number of the inner full clusters packed inside the container and the number of shared clusters at the border, when the default 30° cluster orientation is considered.



**Figure 32.** 4-hexagonal  $H(D)$  clustering result for container size  $D = \{8, 16, 24\}$  and cluster orientation 30° (central cluster position left).

An algebraic expression for calculating the number of inner packed cluster items is as follows:

$$N_{full} = (3D^2 - 24D + 64)/64 \tag{21}$$

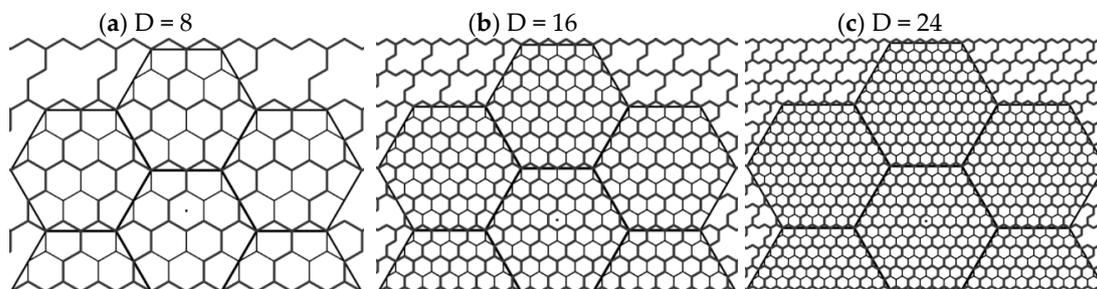
Also, the number of shared clusters at the container border is calculated as:

$$N_{shared} = (3D)/4 \tag{22}$$

**Table 8.** The proposed 4-hexagonal  $H(D)$  clustering model.

Container Size (D)	#Full (Inner) Clusters	#Border (Shared) Clusters
8	1	6
16	7	12
24	19	18
32	37	24
40	61	30

Clustering results from Figure 33 are summarized in Table 9, providing the number of the inner full clusters packed inside the container and the number of shared clusters at the border, when the default 30° cluster orientation is considered.



**Figure 33.** 4-hexagonal  $H(D)$  clustering result for container size  $D = \{8, 16, 24\}$  and cluster orientation 30° (central cluster position up).

**Table 9.** The proposed 4-hexagonal  $H(D)$  clustering model.

Container Size (D)	#Full (Inner) Clusters	#Border (Shared) Clusters
8	1	4
16	8	8
24	21	12
32	40	16
40	65	20

An algebraic expression for calculating the number of inner packed cluster items is as follows:

$$N_{full} = (3D^2 - 16D)/64 \tag{23}$$

Also, the number of shared clusters at the container border is calculated as:

$$N_{shared} = D/2 \tag{24}$$

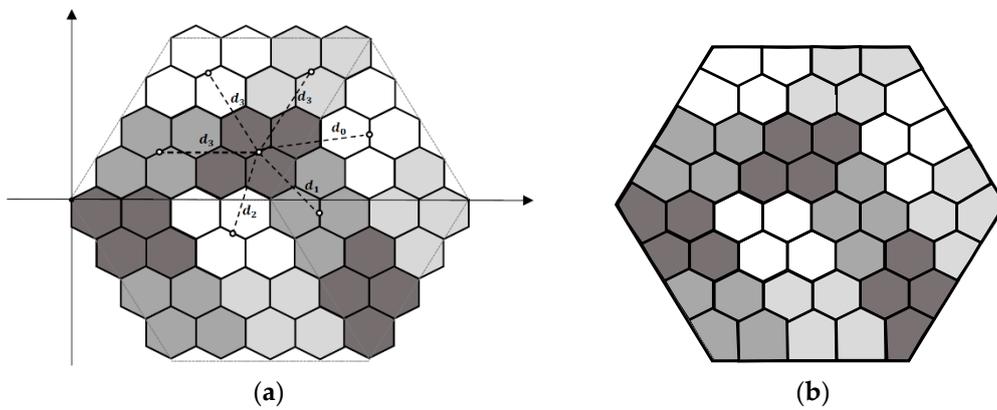
*Cluster Rotation Invariance:* As it can be seen in Figure 28 (for cluster center down), Figure 29 (for cluster center left) and Figure 30 (for cluster center up), the number of clusters  $N_{full}$  and  $N_{shared}$  is not the same for all cluster rotations. The rotation invariance is summarized in Table 10.

**Table 10.** The cluster rotation invariance for  $H(D)$  architectures.

	$H(D)$ Moved Down	$H(D)$ Moved Left	$H(D)$ Moved Up
	$N_{full}, N_{shared}$	$N_{full}, N_{shared}$	$N_{full}, N_{shared}$
$\alpha = 30^\circ$	Formula (19), (20)	Formula (21), (22)	Formula (23), (24)
$\alpha = 90^\circ$	Formula (21), (22)	Formula (23), (24)	Formula (19), (20)
$\alpha = 210^\circ$	Formula (21), (22)	Formula (19), (20)	Formula (19), (20)
$\alpha = 270^\circ$	Formula (19), (20)	Formula (23), (24)	Formula (21), (22)

4.3. The Proposed Non-Uniform Clustering Approach

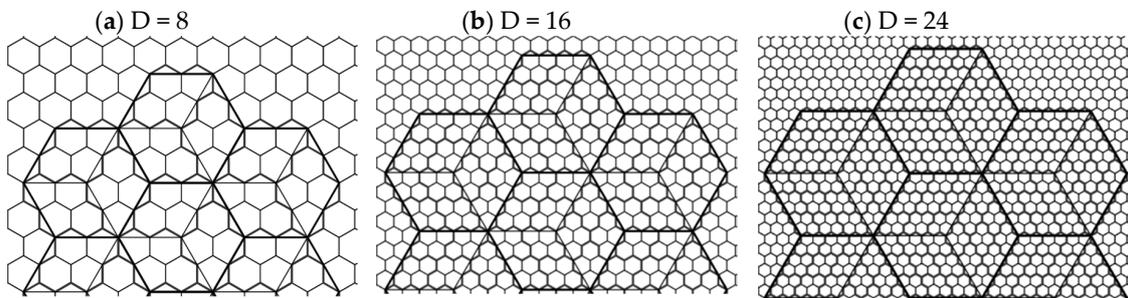
We propose another clustering model that is non-uniform. Inspired by [31], the orientation of the tetrahex clusters is rotated by  $120^\circ$  on each of the container thirds. The non-uniformity is present in the cluster plane, as the distance to all NN clusters is not constant (Figure 34a).



**Figure 34.** The proposed non-uniform clustering. (a) The model with void areas. The cluster NN distances  $d_1, d_2,$  and  $d_3$  are shown; (b) The model without void areas (used in the CMS detector).

We claim that in this proposed architecture, all clusters are entirely contained inside the container with no shared clusters at the border. Again, as already given in Figure 25a, the truncated tetrahexes exist in the clustering procedure, which are now present on three edges of the container. There are voids on the remaining three container edges, in the form of several cut-out triangles. In order to obtain the full benefit of the hexagonal tessellation approach and to eliminate voids inside the container, the model shown in Figure 34b will be used in the future CMS HGCal. Modified tetrahex cluster areas at the container vertex are very close to the real tetrahex ( $\approx 1\%$  difference in average).

An example of packing clusters with the proposed model  $H(D)$  is given in Figure 35.



**Figure 35.** 4-hexagonal  $H(D)$  clustering result in  $120^\circ$  cluster plane for  $D = \{8, 16, 24\}$ .

Clustering results presented in Figure 35 are quantized in Table 11, based on which we derive the corresponding algebraic expressions.

**Table 11.** The proposed 4-hexagonal  $H(D)$  clustering model.

Container Size (D)	#Full (Inner) Clusters	#Border (Shared) Clusters
8	3	0
16	12	0
24	27	0
32	48	0
40	75	0
48	108	0

In this clustering approach, there are no shared clusters at the container border, since they are all contained in the container inner part. Thus, the number of full clusters is calculated with the following formula:

$$N_{full} = 3D^2 / 64 \tag{25}$$

## 5. Evaluation and Results

### 5.1. Summary of the Presented Models

We have shown there are different variations of presented models based on container size, container center position, cluster orientation, or if the void space is allowed or not. We give a brief summary of the presented models in Table 12, together with the models that we have chosen for the comparison.

**Table 12.** The summary of the presented models.

Model	Sub-Type	Sub-Classes	Center Position	Orientations	Chosen Model
CA ( $n = 3k + r$ )	$CA_{even}$ ( $n = 2k$ )	$CA_{even}(0)$ $CA_{even}(1)$ $CA_{even}(2)$	- no movements needed	$\alpha = 30, 90, 150$	$\alpha = 30$
	$CA_{odd}$ ( $n = 2k + 1$ )	$CA_{odd}(0)$ $CA_{odd}(1)$ $CA_{odd}(2)$	- no movements needed	$\alpha = 30, 90, 150$	$\alpha = 30$
VA ( $n = 3k + r$ )	$VA_{even}$ ( $n = 2k$ )	$VA_{even}(0)$ $VA_{even}(1)$ $VA_{even}(2)$	- moved down - moved left	$\alpha = 30, 90, 150, 270, 330$	- moved down $\alpha = 30, 90, 270$
	$VA_{odd}$ ( $n = 2k + 1$ )	$VA_{odd}(0)$ $VA_{odd}(1)$ $VA_{odd}(2)$	- moved down - moved left	$\alpha = 30, 90, 150, 270, 330$	- moved down $\alpha = 30, 90, 270$
$H(D)$ ( $D = 8k$ )	uniform		- moved down - moved left - moved up	$\alpha = 30, 90, 210, 270$	- all positions $\alpha = 30$
	non-uniform	- with voids - no voids			

### 5.2. The Comparison of CA and VA Clustering Architectures

To compare even architectures, we apply Formula (1), Formula (3), Formula (10), Formula (12) to calculate  $N_{full}$  and Formula (4), Formula (9), Formula (11) for  $N_{shared}$ . Figures 36 and 37 present the evaluation of even VA architectures (moved down) compared to CA even, given in the form of the ratio  $N_{full}(VA) / N_{full}(CA)$  and  $N_{shared}(VA) / N_{shared}(CA)$ . Hence, result  $\geq 1$  means the measured parameter is same or larger than CA, while result  $< 1$  means that it is lower. We show that the VA model for  $\alpha = 30^\circ$  is the least efficient in clustering, causing the smallest number of full clusters packed, and the largest number of shared clusters for each container size  $n$ . Also, the VA model for  $\alpha = 90^\circ$  is

the same as the CA in all measurement cases, and this model is the best for container size  $n = 3k$  or  $n = 3k + 2$ . If  $n = 3k + 1$ , the VA model  $\alpha = 270^\circ$  is the most efficient due to the larger number of full clusters than CA and the lowest number of shared clusters at the border.

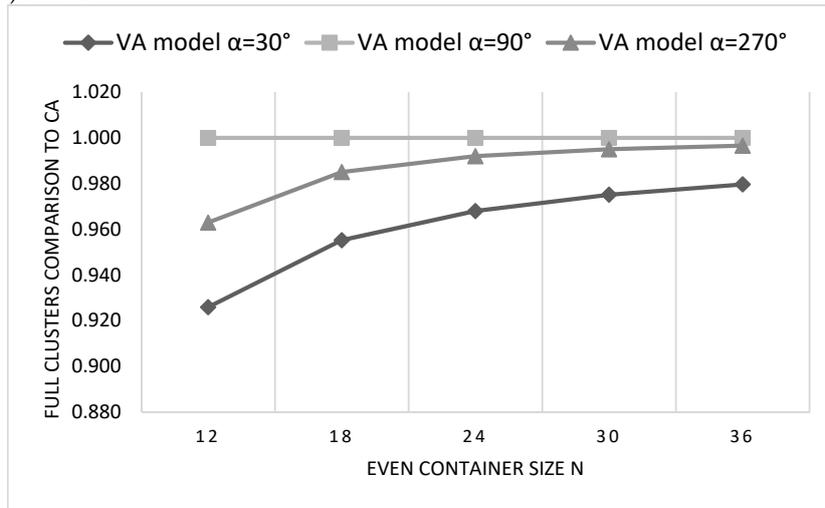
The comparison in Figures 36 and 37 is based on even containers, where a single container type  $N$  is present independent of how many rings of containers there are tessellated in the clustering plane. However, we already concluded that the number of different container types is larger for odd than for even container sizes. Hence, there will be several container configurations in a single tessellation ring depending on the container size, and we show them in Figure 38. We approximate the number of full and shared clusters in a single container ring with the formulas presented, to obtain a fair comparison of architectures.

To compare odd architectures, we apply configurations based on Formula (6), Formula (7), Formula (14), Formula (15) to calculate  $N_{full}$  and Formula (8), Formula (9), Formula (16), Formula (17), Formula (18) for  $N_{shared}$ . We can see in Figures 39 and 40 that the VA model for  $\alpha = 30^\circ$  is the worst among odd clustering architectures since there are the most shared clusters in the container ring for the lowest number of inner full clusters. On the other hand, VA model  $\alpha = 270^\circ$  is the most efficient in all cases for  $n = 3k$  and  $n = 3k + 2$ , while for  $n = 3k + 1$  the VA model  $\alpha = 90^\circ$  is the best, as there are more full clusters than CA in the container ring.

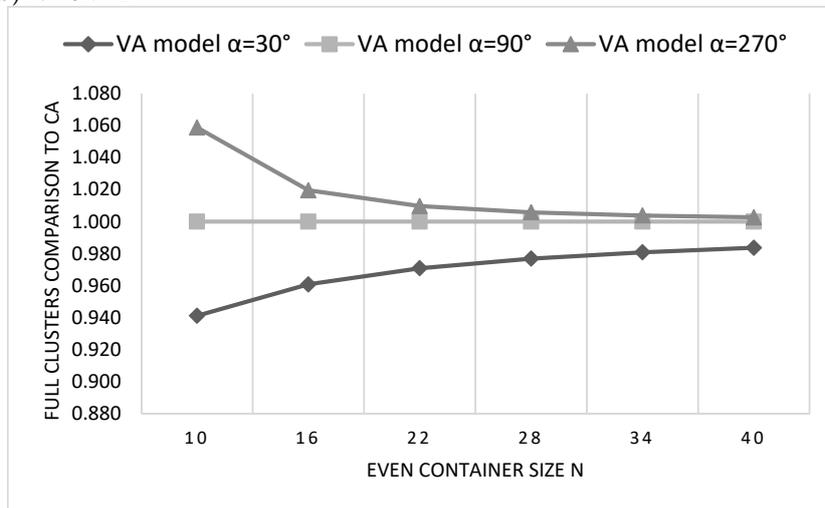
It is shown in Figure 39a that the odd CA is more efficient than any odd VA model for  $n = 3k$ , having the largest number of full clusters in the container ring. If we compare VA models, VA  $\alpha = 270^\circ$  is the best, being closest to CA efficiency. However, as shown in Figure 40a, VA models  $\alpha = 90^\circ$  and  $\alpha = 270^\circ$  are better to use than CA if the goal is not to maximize the number of clusters packed, but to minimize the number of shared clusters.

In addition, all odd VA models are less efficient than CA (Figure 40b), causing larger cluster sharing, even though the number of full clusters is enhanced in the VA case (Figure 39b).

(a)  $N = 3k$



(b)  $N = 3k + 1$



(c)  $n = 3k + 2$

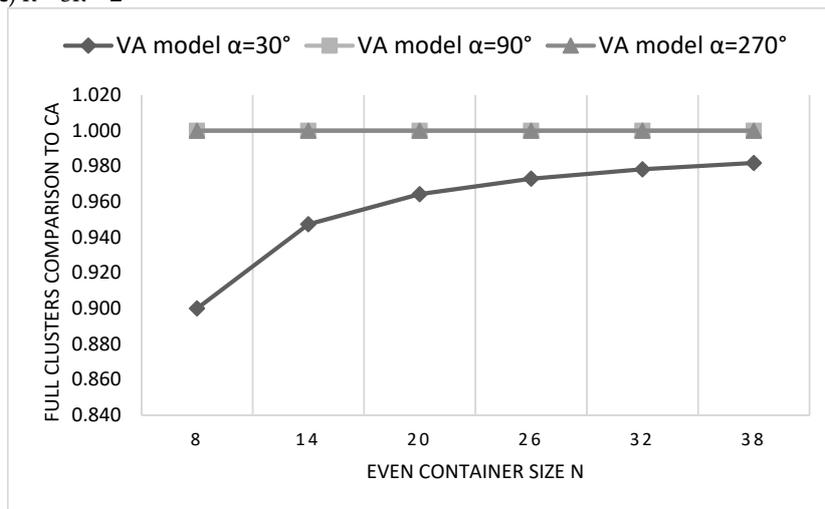


Figure 36. The number of full clusters compared for CA and VA even models.

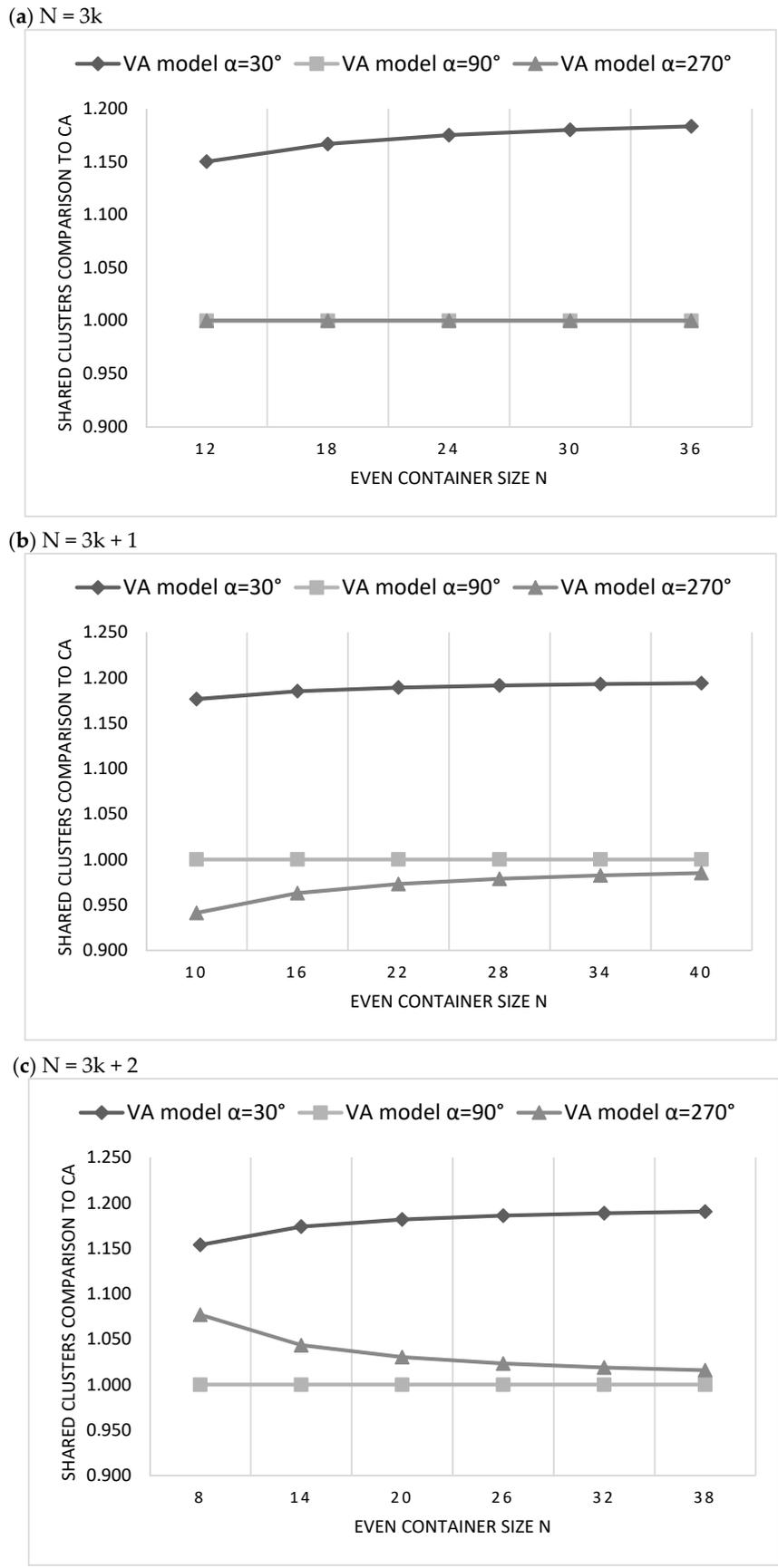
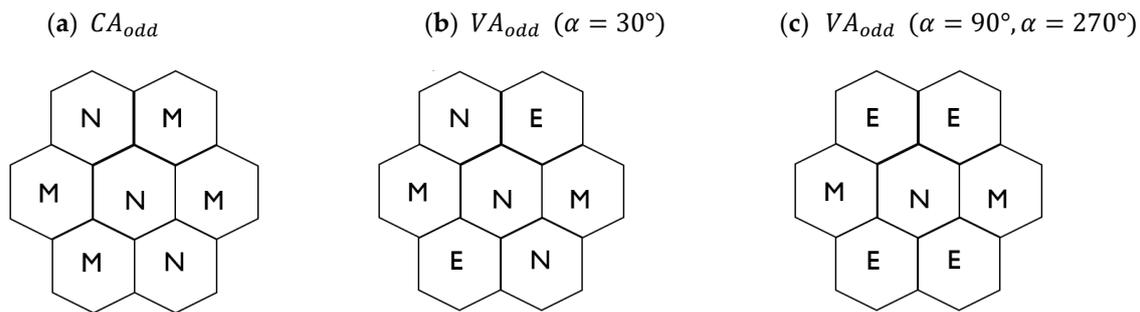


Figure 37. The number of shared clusters compared for CA and VA even models.



**Figure 38.** Clustering configurations in a single ring of container tessellation. (a)  $3N + 4M$ ; (b)  $3N + 2M + 2E$ ; (c)  $N + 2M + 4E$ .

### 5.3. The Evaluation of the Proposed $H(D)$ Clustering Model

To compare the proposed  $H(D)$  models for uniform and non-uniform clustering, we use Formula (19), Formula (21), Formula (23), Formula (25) for  $N_{full}$  and Formula (20), Formula (21), Formula (24) for  $N_{shared}$ . The efficiency result is shown in Figure 41. We use only the default cluster orientation  $\alpha = 30^\circ$  for the uniform clustering models (moved left, moved down, and moved up). The maximized number of full clusters or packed items is obtained for uniform  $H(D)$  with the cluster plane moved down. Also, it has the lowest number of shared clusters among uniform architectures. On the other hand,  $H(D)$  moved left the least efficient, due to the chosen cluster angle. Namely, since sharing clusters are present at every edge of the container in  $H(D)$  moved left and  $\alpha = 30^\circ$ , it has the lowest occupancy. In order to minimize the number of shared clusters for each uniform model, we need to choose rotations  $\alpha = 30^\circ, 90^\circ, 210^\circ$  to have shared clusters only at two edges of the container and higher occupancy for architectures moved down, up and left respectively.

Compared to clustering with the existing CA and VA models, these uniform  $H(D)$  architectures are more efficient. Namely, in terms of shared clusters, sharing is present at each of six edges of the container with the existing models, while here clusters are shared only at some of the container edges, where sharing on two edges is the minimal solution which can be obtained. The proposed non-uniform  $H(D)$  model is the most efficient of all since the number of packed items in the container is maximized, and it requires no cluster sharing.

The usual goal of solving the packing problem is to minimize the void area in the container, so we define occupancy as the percentage of the container filled with polyhex clusters. It is evaluated by using the formulas for  $N_{full}$ . First, we calculate the area of the container based on total number of hexagonal cells inside ( $N_{total} = 4 * N_{full} = 3D^2/16$  from Formula (25)), and then we calculate the number of “used” or clustered cells  $N_{clustered\ cells} = 4 * N_{full}$  based on Formula (19), Formula (21) and Formula (23). The occupancy is expressed as ratio between the two:  $N_{clustered\ cells}/N_{total}$ .

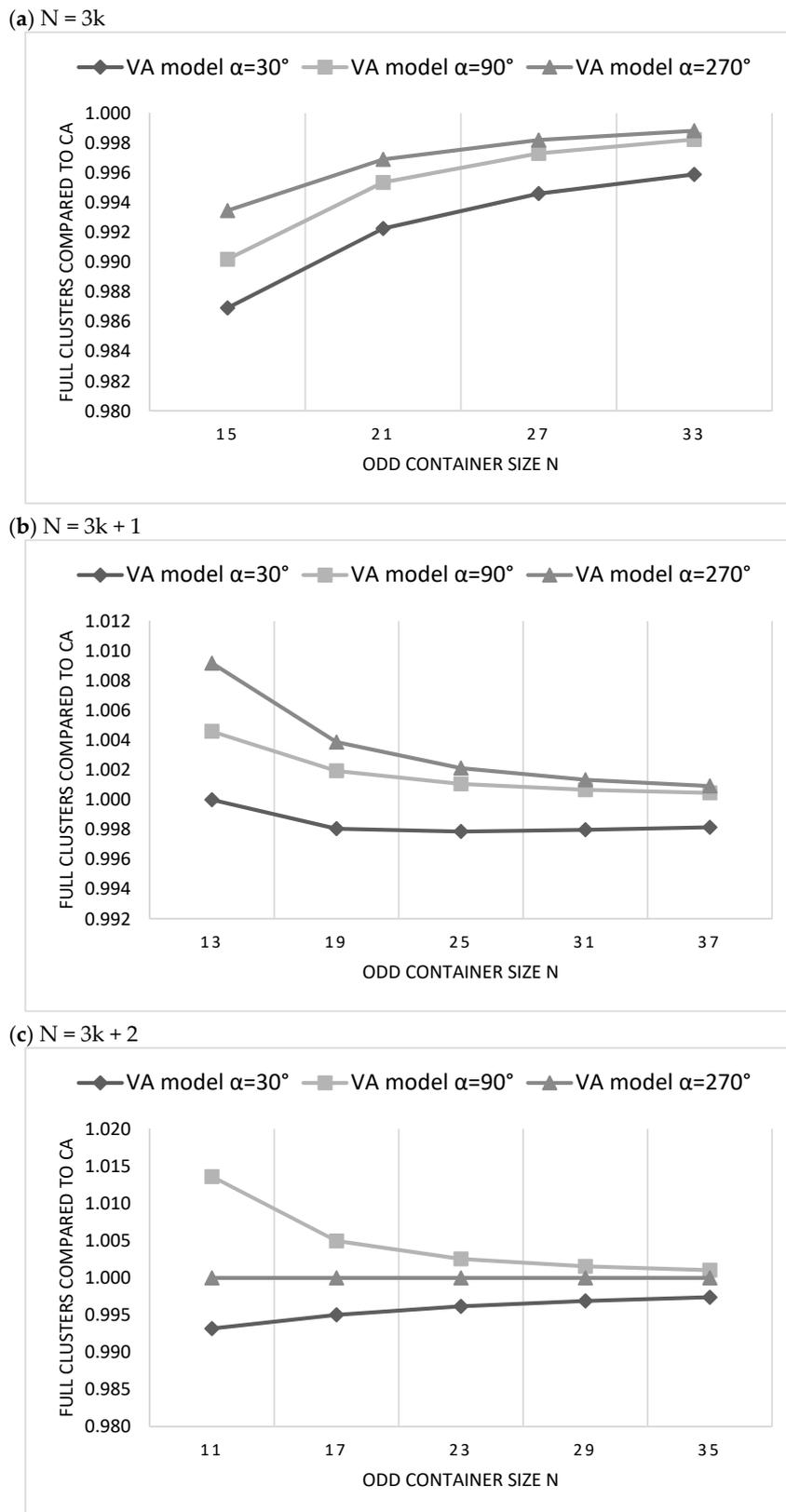


Figure 39. The number of full clusters compared for CA and VA odd models.

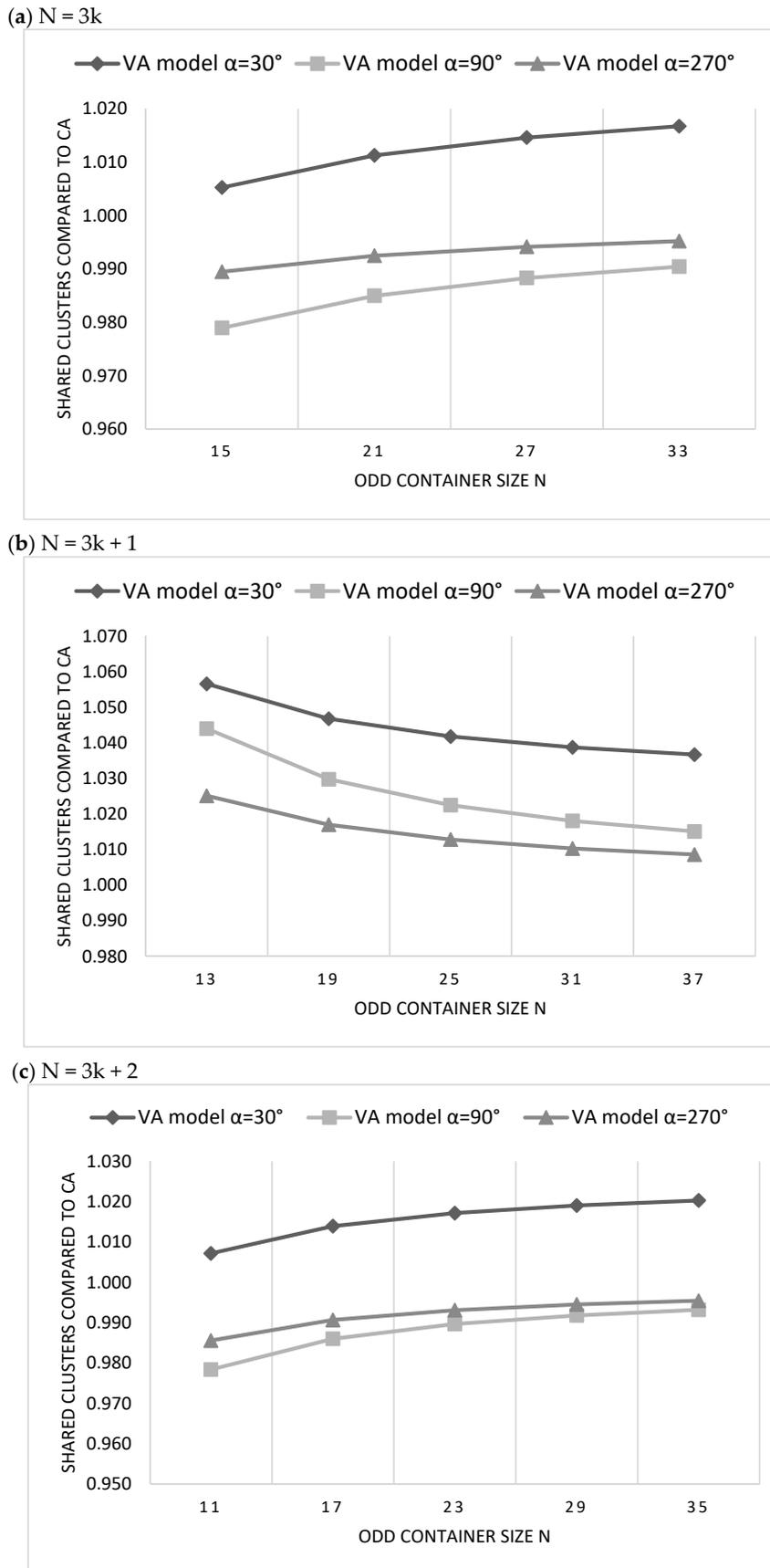
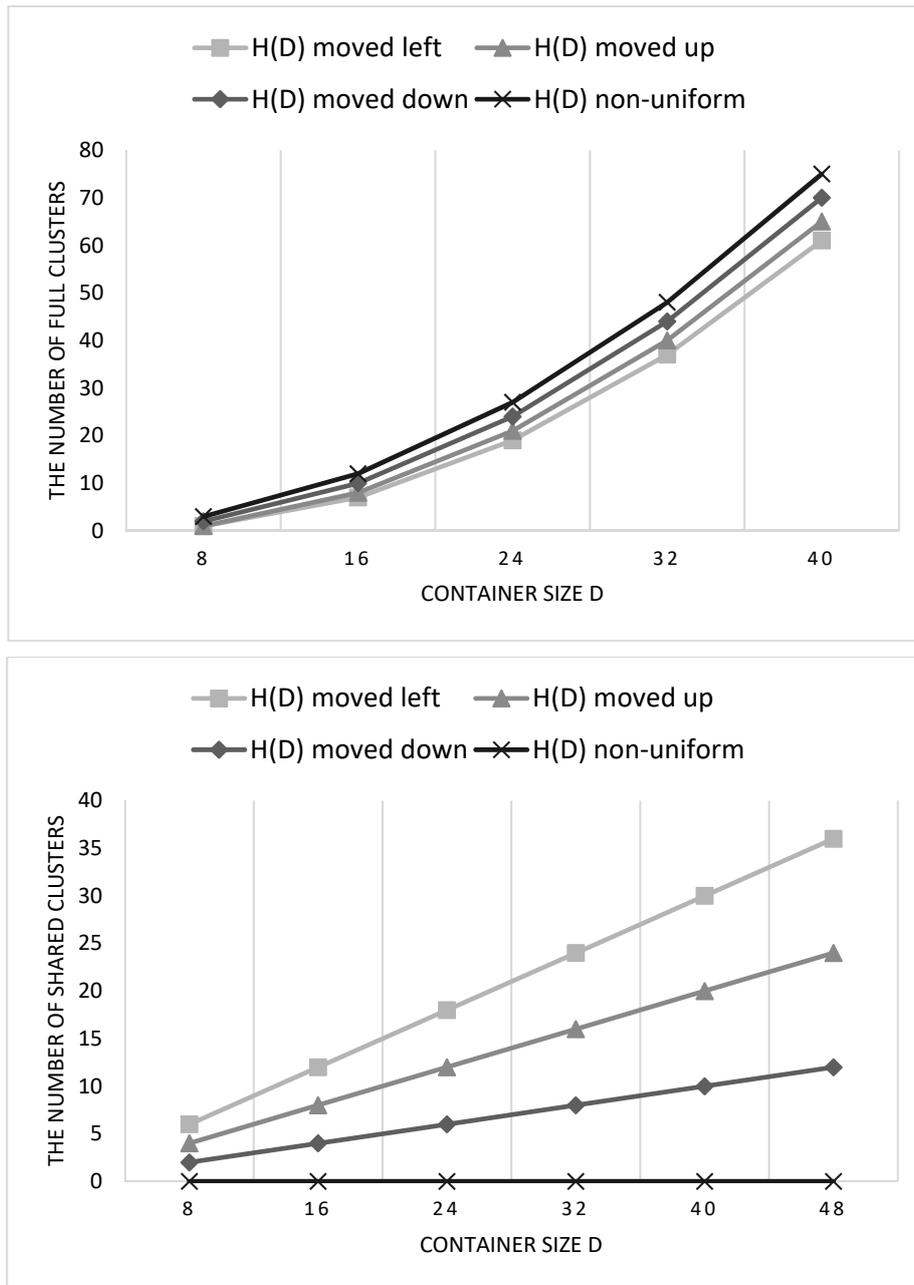


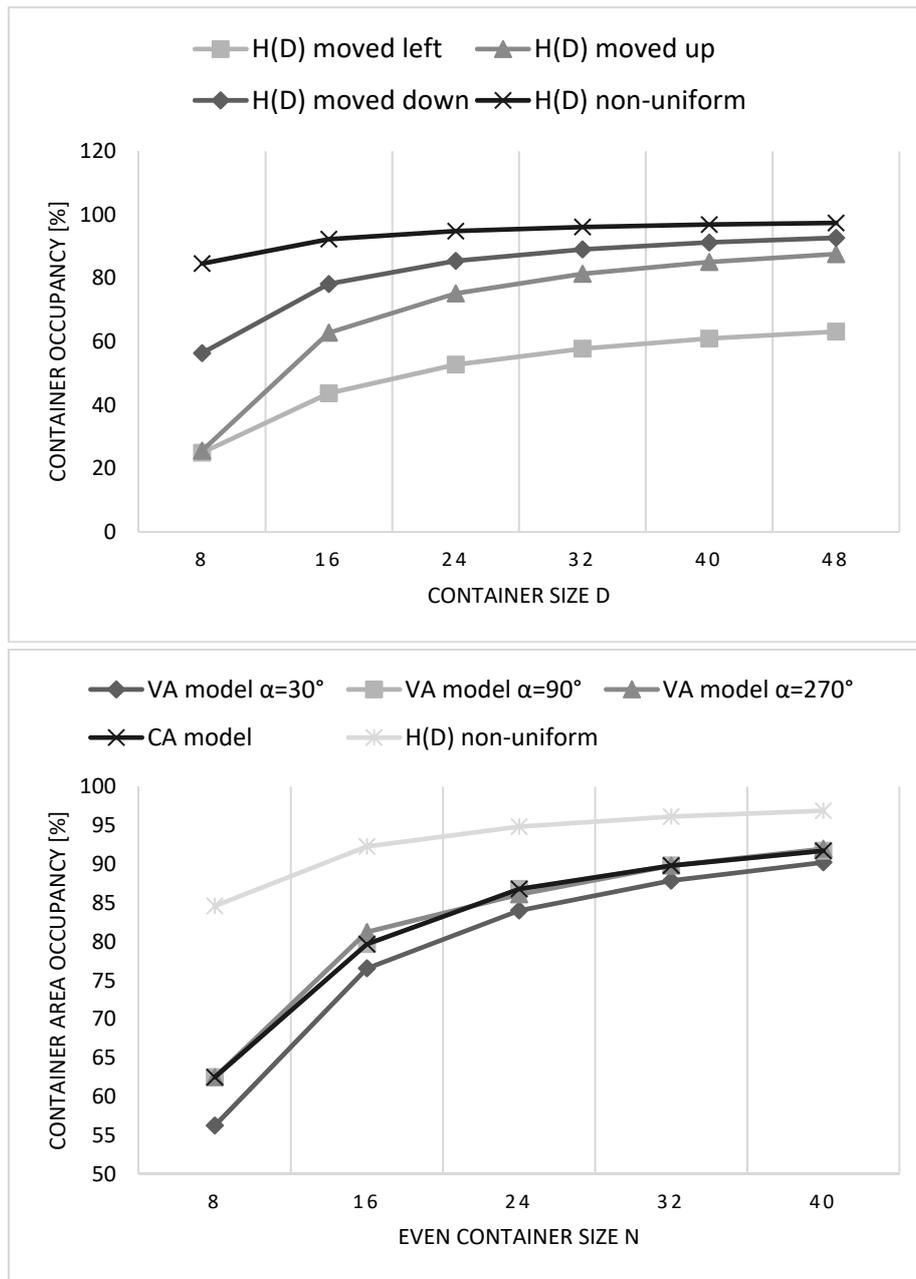
Figure 40. The number of shared clusters compared for CA and VA odd models.



**Figure 41.** The number of full and shared clusters compared for the proposed  $H(D)$  models.

We can see in Figure 42 that the occupancy of the non-uniform  $H(D)$  container model is maximal, making it the most efficient than every other architecture. Since sharing is present at every container edge with the existing models, these shared clusters are unpacked and cause the void areas. There is no cluster sharing in the non-uniform  $H(D)$  structure, and all clusters are completely contained inside the container border, so the void area is minimal in this type of container. The additional enhancement is obtained with the model applied for the real CMS detector, so that container occupancy is 100% (Figure 34b).

It is to note that the efficiency of some uniform  $H(D)$  architectures (moved left and moved up) is less than the existing models ( $\approx 48\%$  and  $\approx 66\%$  on average).



**Figure 42.** The container occupancy comparison between H(D) models and comparison with the existing VA and CA architectures.

## 6. Limitations of the Study

There are a few limitations to the current study. First, clusters are assumed to be homogeneous in area and shape and composed of the same components forming the uniform cluster plane. Although several different polyhex types can be constructed, only a single cluster shape is considered in this paper, when four hexagonal cells are merged together and generating 4-hexagonal or tetrahex clusters and in our application-driven context of using “bee” tetrahex shape. We examined the cluster packing of this specific form, which is most common in engineering, and it is suitable for forming sensor “trigger cells” in CMS detector. Other polyhex structures could have also been formed and analyzed to obtain a more generic study.

Besides those general features of the current research, there are some other limitations related to specific sections in the paper. Concerning the evaluation section, for instance, when we compare the

clustering with the two existing approaches (CA and VA), we consider only VA architectures moved down. This is done not to add additional complexity to the study, and one can easily derive the results for VA moved left, with cluster rotation angles  $30^\circ$ ,  $150^\circ$ , and  $330^\circ$ . Also, when comparing the number of full and shared clusters for  $CA_{odd}$  and  $VA_{odd}$ , we consider a ring of tessellated containers with packed clusters. We took only a single ring, and one could increase the number of rings depending on the application. This can influence the conclusions from the obtained results, due to the increased number of container types with these architectures. Also, the ring configurations for VA are derived for the default case (moved down,  $\alpha = 30, 90, 270$ ), and they are not the same when cluster rotation angle is varied. Similarly, when comparing the proposed  $H(D)$  models, we concentrate only on the default cluster orientation  $\alpha = 30^\circ$ . The additional factor of the varied cluster rotation may influence on the obtained results.

Finally, the limitation of the proposed  $H(D)$  models is that the cluster area is not constant for all packed clusters, since some clusters at the container border will be smaller in the area caused by the two truncated hexagon pieces. However, this is easily prevailed with the deformed clusters at the container border, so that all inner clusters are very similar in the area. The only compromise which can be considered a limitation of the proposed  $H(D)$  non-uniform model is the non-uniformity of the cluster plane. Since the distance from each central cluster to NN clusters is not constant, it could slightly complicate the NN finder algorithm used in the cluster grouping procedure.

## 7. Conclusions

In this paper, we revisit a problem of packing symmetric “bee” tetrahex clusters in a regular hexagonal container. We start from existing container models from the literature (CA and VA). We apply these models in a new context of clustering the inner hexagonal cells in polyhex clusters and packing them in the container. We take a single ring of tessellated containers with inner packed clusters, and we formally describe the geometrical and algebraic properties by using the derived formulas. We analyze the cluster plane and derive proofs of the geometrical structures. We propose a new vertex-aligned  $H(D)$  model, which is more efficient than the existing approaches.

The packing efficiency is evaluated in terms of five criteria: the total number of the inner packed items (full clusters), the total number of shared clusters at the container border, the number of container types, the cluster uniformity, and whether clusters are the same in area. Since the usual goal, when solving the packing problem is to minimize the void area, we derive the container occupancy formula to evaluate the efficiency of the proposed model. The following conclusions are obtained from the study based on the desired objectives:

- Maximal number of packed items

When comparing the even  $CA_{even}$  and  $VA_{even}$  architectures (moved down,  $\alpha = 30^\circ, 90^\circ, 270^\circ$ ), they are all very similar, independent on the container size. The container occupancy is, on average, 82%, while VA ( $\alpha = 30^\circ$ ) occupancy is the lowest with  $\approx 79\%$ . Similarly, all the existing odd architectures  $CA_{odd}$  and  $VA_{odd}$  are almost the same, with the average occupancy in a single ring  $\approx 78\%$ . Concerning the proposed  $H(D)$  models ( $\alpha = 30^\circ$ ),  $H(D)$  moved down is the best, having the average packed area  $\approx 13\%$  and  $\approx 32\%$  larger than the moved up and the moved left architectures, respectively. It is on average  $\approx 3\%$  better than the existing models.  $H(D)$  architecture moved left the least efficient, with occupancy  $\approx 50\%$ . The proposed non-uniform  $H(D)$  is the most efficient, with the maximized packed area inside the container (the occupancy is  $\approx 93.7\%$  on average).

- Minimal number of shared clusters

The existing architectures have larger cluster sharing at the container border, i.e., sharing is present on each of the six container edges, which increases the non-used or void container space. These shared clusters only take place in the container and prevent us from packing the whole cluster inside the container border. The importance of the proposed  $H(D)$  models is that their efficiency

higher. For example, with the uniform  $H(D)$  models, we can obtain reduced cluster sharing on only two container edges. Hence, communication is reduced between boards that are processing data from each of the neighboring containers separately. This is important in the real CMS detector design so that a single sensor module (hexagonal container) has all “trigger cells” (tetrahex clusters) packed inside. Also, the non-uniform  $H(D)$  is the most efficient in this sense, as it allows no cluster sharing at all, so the communication between neighboring containers is avoided. Still, voids are present in the container (the occupancy is  $\approx 93.7\%$ , so the non-used area is  $\approx 6.3\%$ ), and they are eliminated by deforming the border clusters, such as in the real CMS detector application.

- Single container type

The existing architectures  $CA_{even}$  and  $VA_{even}$  provide a single container type when a ring of containers with inner packed clusters is tessellated. Clustering with architectures  $CA_{odd}$  and  $VA_{odd}$  causes a larger number of different container types making them less efficient. Also, a single container type is allowed with the proposed  $H(D)$  architectures.

- Uniform clustering model

The uniform clustering is provided by the existing architectures CA and VA and the proposed uniform  $H(D)$  models. However, in the non-uniform  $H(D)$  model, the orientation angle of the inner clusters is not unique, as well as the distance from each central cluster to its NNs.

- All clusters identical in the area

Unlike the existing architectures, the proposed  $H(D)$  models do not allow all clusters to be the same in area. However, the compromise is negligible as the deformed clusters are very close in the area to a real “bee” tetrahex ( $\approx 9.1\%$  if the voids are allowed in the container and  $\approx 1\%$  if the border clusters are deformed to eliminate void areas).

The clustering scheme with the proposed non-uniform  $H(D)$  model is the most efficient to solve the polyhex packing problem, since it prevails in four out of five evaluation criteria. Larger container occupancy is provided compared to the existing architectures, but what is more important, sharing is completely omitted at the container border. The only compromise is the non-uniformity itself, since the criteria of all clusters having the same area is easily overcome with the deformed border tetrahex clusters. Also, there are no voids in the container this way, as the resulting occupancy is 100%. All these qualities enable the direct application of our study, i.e., the proposed cluster packing scheme of “bee” tetrahex clusters that will be used to create “trigger cells” in the new CMS HGCAL detector at CERN.

Based on the references cited, we believe that our work has a number of indirect practical applications, and we provide a view of the future use of our proposed model. For example, polyhex tiling is considered useful in robotics, and using tetrahex with an objective of maximizing the area coverage is needed for a coverage path-planning technique in a self-reconfigurable floor cleaning robot. Our model can be applied for “bee” polyhex-based morphology, but if the robot movement is limited on the targeted hexagonal ROI. Since the occupancy is the largest, and there are no void areas, the maximal area coverage can be obtained. Also, our model can be used when the polygonal area is covered with wireless sensor devices that are hierarchically clustered in “bee” polyhexes to obtain the increased energy efficiency. It would be interesting to examine the position of the cluster base stations since “bee” cluster does not have the central cell in one of the centers of the contained hexagonal cells. Our model can also be extended to consider other polyhex types and container shapes. Hence, a more general study can be obtained, providing a framework for packing different types of (not necessarily regular) convex polygons.

**Author Contributions:** All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

CA	Center Aligned
CERN	European Laboratory for Particle Physics
CMS	Compact Muon Solenoid
HGCAL	High Granularity Calorimeter
LHC	Large Hadron Collider
NN	Nearest Neighbor
ROI	Region of Interest
VA	Vertex Aligned

## References

- Romanova, T.; Pankratov, A.; Litvinchev, I.; Pankratova, Y.; Urniaieva, I. Optimized Packing Clusters of Objects in a Rectangular Container. *Math. Probl. Eng.* **2019**, *2019*, 4136430. Available online: <https://www.hindawi.com/journals/mpe/2019/4136430/> (accessed on 13 December 2019). [[CrossRef](#)]
- Litvinchev, I.; Infante, L.; Ozuna, L. Packing circular-like objects in a rectangular container. *Comput. Syst. Sci. Int.* **2015**, *54*, 259–267. [[CrossRef](#)]
- Torres-Escobar, R.; Marmolejo-Saucedo, J.A.; Litvinchev, I.; Vasant, P. Monkey Algorithm for Packing Circles with Binary Variables. *Intell. Comput. Optim. Cham.* **2019**, 547–559. [[CrossRef](#)]
- Kampas, F.J.; Castillo, I.; Pintér, J.D. Optimized ellipse packings in regular polygons. *Optim. Lett.* **2019**, *13*, 1583–1613. [[CrossRef](#)]
- Kampas, F.J.; Pinter, J.D.; Castillo, I. Packing Ovals in Optimized Regular Polygons. *arXiv* **2019**, arXiv:1901.07056. [[CrossRef](#)]
- Pankratov, A.; Romanova, T.; Litvinchev, I. Packing ellipses in an optimized convex polygon. *J. Glob. Optim.* **2019**, *75*, 495–522. [[CrossRef](#)]
- López, C.O.; Beasley, J.E. Packing a fixed number of identical circles in a circular container with circular prohibited areas. *Optim. Lett.* **2019**, *13*, 1449–1468. [[CrossRef](#)]
- Bennell, J.; Scheithauer, G.; Stoyan, Y.; Romanova, T.; Pankratov, A. Optimal clustering of a pair of irregular objects. *J. Glob. Optim.* **2015**, *61*, 497–524. [[CrossRef](#)]
- Peralta, J.; Andretta, M.; Oliveira, J.F. Packing Circles and Irregular Polygons using Separation Lines. In Proceedings of the 7th International Conference on Operations Research and Enterprise Systems, Funchal, Portugal, 24–26 January 2018; pp. 71–77. [[CrossRef](#)]
- Galiev, S.; Lisafina, M. Numerical optimization method for packing regular convex polygons. *Comput. Math. Math. Phys.* **2016**, *56*, 1402–1413. [[CrossRef](#)]
- Prvan, M.; Ožegović, J.; Mišura, A.B. On Calculating the Packing Efficiency for Embedding Hexagonal and Dodecagonal Sensors in a Circular Container. *Math. Probl. Eng.* **2019**, *2019*, 9624751. Available online: <https://www.hindawi.com/journals/mpe/2019/9624751/cta/> (accessed on 10 December 2019). [[CrossRef](#)]
- Wang, D.; Lin, L.; Xu, L. A study of subdividing hexagon-clustered WSN for power saving: Analysis and simulation. *Ad Hoc Netw.* **2011**, *9*, 1302–1311. [[CrossRef](#)]
- Stoyan, Y.; Yaskov, G. Packing equal circles into a circle with circular prohibited areas. *Int. J. Comput. Math.* **2012**, *89*, 1355–1369. [[CrossRef](#)]
- Galiev, S.I.; Lisafina, M.S. Linear models for the approximate solution of the problem of packing equal circles into a given domain. *Eur. J. Oper. Res.* **2013**, *230*, 505–514. [[CrossRef](#)]
- Toledo, F.M.B.; Carravilla, M.A.; Ribeiro, C.; Oliveira, J.F.; Gomes, A.M. The Dotted-Board Model: A new MIP model for nesting irregular shapes. *Int. J. Prod. Econ.* **2013**, *145*, 478–487. [[CrossRef](#)]
- Kazakov, A.L.; Lempert, A.A.; Nguyen, H.L. The Problem of the Optimal Packing of the Equal Circles for Special Non-Euclidean Metric. In *International Conference on Analysis of Images, Social Networks and Texts*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 58–68.
- López, C.O.; Beasley, J.E. Packing unequal rectangles and squares in a fixed size circular container using formulation space search. *arXiv* **2018**, arXiv:1802.07519.
- Ekanayake, D.B.; Ranpatidewage, M.M.; LaFountain, D.J. Optimal Packings for Filled Rings of Circles. *Appl. Math.* **2020**, *65*, 1–22. [[CrossRef](#)]

19. Lintzmayer, C.N.; Miyazawa, F.K.; Xavier, E.C. Online Circle and Sphere Packing. *arXiv* **2017**, arXiv:1708.08906.
20. Coffin, S.T. *The Puzzling World of Polyhedral Dissections*; Oxford University Press: Oxford, UK, 1991.
21. Parween, R.; Shi, Y.; Parasuraman, K.; Vengadesh, A.; Sivanantham, V.; Ghanta, S.; Mohan, R.E. Modeling and Analysis of hHoneycomb—A Polyhex Inspired Reconfigurable Tiling Robot. *Energies* **2019**, *12*, 2517. [[CrossRef](#)]
22. Luo, J.; Zhang, W.; Su, J.; Xiang, F. Hexagonal Convolutional Neural Networks for Hexagonal Grids. *IEEE Access* **2019**, *7*, 142738–142749. [[CrossRef](#)]
23. Uher, V.; Gajdoš, P.; Snášel, V.; Lai, Y.-C.; Radecký, M. Hierarchical Hexagonal Clustering and Indexing. *Symmetry* **2019**, *11*, 731. [[CrossRef](#)]
24. Sauvan, J.-B. Concepts and design of the CMS high granularity calorimeter Level-1 trigger. *J. Phys. Conf. Ser.* **2017**, *928*, 012026. [[CrossRef](#)]
25. Collaboration, C.M.S. The Phase-2 Upgrade of the CMS Endcap Calorimeter. CERN Document Server, 21 November 2017. Available online: <https://cds.cern.ch/record/2293646> (accessed on 10 December 2019).
26. Maier, A.A. Sensors for the CMS High Granularity Calorimeter. *J. Instrum.* **2017**, *12*, C06030. [[CrossRef](#)]
27. Yang, W. *Maximal and Minimal Polyhexes*; University of Wisconsin-Madison: Madison, WI, USA, 2002.
28. Mahdavi-Amiri, A.; Harrison, E.; Samavati, F. Hexagonal connectivity maps for Digital Earth. *Int. J. Digit. Earth* **2015**, *8*, 750–769. [[CrossRef](#)]
29. Holub, P.; Miller, M.; Pérez-Rosés, H.; Ryan, J. Degree diameter problem on honeycomb networks. *Discrete Appl. Math.* **2014**, *179*, 139–151. [[CrossRef](#)]
30. Tong, X.; Ben, J.; Wang, Y. A new effective Hexagonal Discrete Global Grid System: Hexagonal quad balanced structure. In Proceedings of the 2010 18th International Conference on Geoinformatics, Beijing, China, 18–20 June 2010; pp. 1–6. [[CrossRef](#)]
31. Van de Ville, D.; Blu, T.; Unser, M.; Philips, W.; Lemahieu, I.; Van de Walle, R. Hex-Splines: A Novel Spline Family for Hexagonal Lattices. *IEEE Trans. Image Process.* **2004**, *13*, 758–772. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).