Anatolian Journal of Computer Sciences

Volume: 4 No: 1 2019 © Anatolian Science
pp: 38-46 ISSN:2548-1304

Perfect Matching of Fractal Honeycomb Meshes

Burhan SELÇUK[1], *Ayşe Nur ALTINTAŞ TANKÜL*[1]
[1]*Computer Engineering, Karabuk, Turkey (bselcuk@karabuk.edu.tr aysenuraltintas@karabuk.edu.tr)*

Received: Dec.19, 2019
Accepted: Jan.3, 2019
Published: Jun. 1, 2019

***Abstract*:** In this paper, we investigated Hamiltonian properties of fractal honeycomb meshes which are created in two different ways using 2-bit gray code. We presented the structure of honeycomb meshes and examined the fractal properties of them and got perfect matching of labeling of nodes in Fractal Honeycomb Meshes for any dimension.

Keywords: *Fractal honeycomb meshes, Interconnection network, Hamilton graph, Perfect matching, Gray code.*

1. Introduction

Network topology is an illustration of nodes and their connections. There are different types of network topologies and bus, ring, star, tree, mesh, tori and hypercube topologies are the most commonly known network topologies. In this paper, we used honeycomb pattern to construct network topology in fractal-like structure with two variants of honeycomb meshes and examined their Hamilton properties.

Honeycomb pattern is inspired from nature and it has rigid, strong and lightweight structure. The armadillos' shells, marine skeletons, insect eyes and, certainly, the wax cells built by bees are just a few examples of nature's honeycombs. Scientist have known the strengths of honeycomb from before, studied on its structural properties from the early 20th century (Hales T. C., 2001) and recent years it is used in different applications such as cellular phone station positioning (Nocetti F. G., Stojmenovic I., Zhang J., 2002), design and use of multiprocessor interconnection networks(Carle J., Myoupo J. F., Seme D., 1999; Manuel P., Rajan B., Rajasingh I., M C. M., 2008), computer graphics (Lester L. N., Sandor J., 1985), modelling chemical structures in chemistry (Rajan B., William A., Grigorious C., Stephen S., 2012), component structure for satellites (Boudjemai A., Amri R., Mankour A., Salem H., Bouanane M. H., Boutchicha D., 2012) and even in tissue engineering for heart (Engelmary G. C., Cheng M., Bettinger C. J., Borenstein J. T., Langer R., Freed L. E., 2008). Many authors have examined Hamilton properties in Honeycomb torus and meshes (Cho, H. J., Hsu, L. Y., 2003, Yang, X., Evans, D. J., Lai, H., Megson, G. M. 2004). Honeycomb network has fractal features and can be used for representing complex networks.

Fractal has come from the word Latin ***frâctus***, which means shattered or broken. A Polish mathematician Benoit Mandelbrot used the term "fractal" in 1975 for the first time (Mandelbrot B., 2004). Patterns constructed with proportional reduction or enlargement of a shape are called fractal. One characteristic of the fractal is that the pattern in a small part is the same as the pattern in the entire shape. Even if it is repeated and seems like regular, it can be seen any kind of irregular or random things. The discovery of fractal geometry made it possible to mathematically investigate coarse irregularities in nature. In nature, the branching of tracheal tubes, the veins in a hand, a cumulus cloud, water swirling, the leaves in trees, the DNA molecule or the oxygen molecule are examples for fractal formation (Kluge T., 2000). Fractals are better to describe the real world than a traditional physics and mathematics, so, it is more and more used for the applications in art and science. Art (Joye Y., 2005), astronomy, computer science (Keller J., Chen S., 1989; Karci A., Selçuk B., 2014), fluid mechanics, telecommunications, surface physics, medicine are some of the areas where fractal is used (Kluge T., 2000).

A graph $G = (V, E)$ is geneally used to represent an interconnection networks where V is a set of nodes and E is a set of edges. A honeycomb graph has been used for constructing interconnection networks. It can be represented recursively. Each node in honeycomb graph can be labeled with 2-bit gray code and successive codewords are different from each other only in a one bit.

There are different types of graph construct from fractals (Meier J., Reiter C. A., 1996; Warchalowski W., Krawczyk M. J., 2017; Fiala J., Hubička J., Long Y., Nešetřil J., 2017; Jaggard D. L., Bedrosian S. D., 1987; Montiel M. E., Aguado A. S., Zaluska E., 1995; Brown J. I., Hickman C. A., Nowakowski R. J., 2003; Ejov V., Filar J. A., Lucas S. K., Zograf P., 2007). Generating of hierarchical scale-free graphs from fractals is shown by Komjathy et al. in (Science N., Phenomena C., Komjáthy J., Simon K., 2011). In this paper, we constructed a honeycomb network topology from fractal structure and examined the Hamiltonian-like properties of this graph.

We defined 2-bit gray code in the following table that is used in the rest of the paper;

Table 1. 2-bit Gray Code.

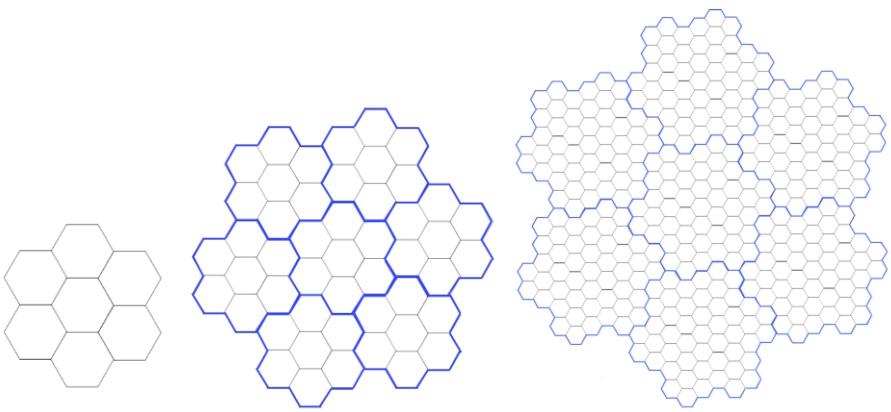
A	B	C	D
00	01	11	10

Here, we investigated Hamilton properties of Fractal Honeycomb Meshes. The paper is organized as follows: Section 2 informs about variants of fractal honeycomb meshes. Hamilton properties of Fractal Honeycomb meshes are explained and two recursive algorithms are given in Section 3. These recursive algorithms are used for labeling the nodes of mesh structure is given in Section 2. In section 4, the conclusion of this work is stated.

2. Variants of Fractal Honeycomb Meshes

Honeycomb Networks can be used for hierarchical fractal structure. In this section, we consider two variants of Fractal Honeycomb Meshes called FHC and FHCC. Both FHC and FHCC are suitable for creating this structure.

Case 1: Constructing FHC hierarchical fractal, six hexagons are placed around central one for 1st order structure. For second order, around 1st order structure, six duplications are placed and similarly for third order, around 2nd order structure, six duplications are placed.



(a) (b) (c)

Figure 1. Fractal HC **(a)** 1st order **(b)** 2nd order **(c)**3rd order

Case 2: Constructing FHCC hierarchical fractal, four hexagons are placed without overlapping to form 1st order structure, two of these hexagons have two neighbor and other two hexagons have three neighbors. For second order, four 1st order structure FHCC and for third order, 2nd order structures are placed and similarly.

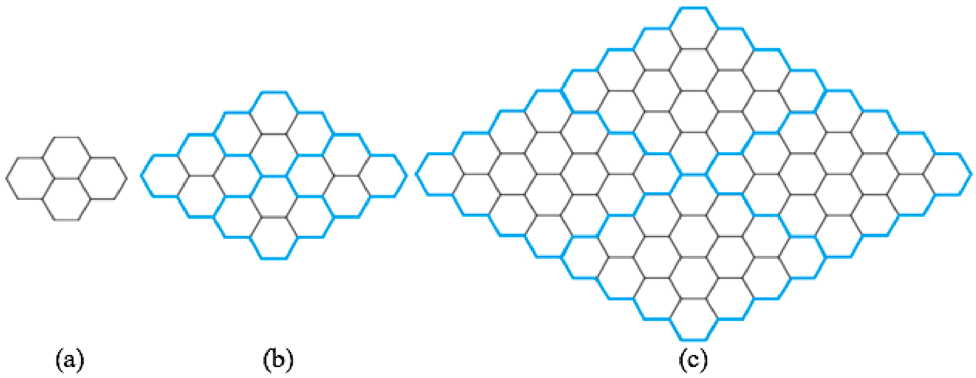


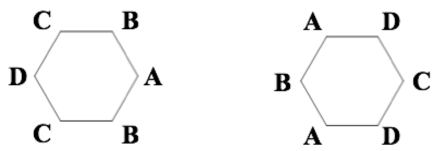
Figure 2. Fractal HCC (a) 1st order (b) 2nd order (c)3rd order

3. Hamiltonian Properties of Fractal Honeycomb Meshes

3.1. FHC Networks

In this subsection, we introduce the construction of $\mathrm{FHC(n)}$.The Fractal Honeycomb network $\mathrm{FHC}(0)$ is a hexagon. As shown in figure 3, we find two different labeling of FHC for $n = 0$ choosing the starting node is D or B;

$$\mathrm{FHC}(0) = D \to C \to B \to A \to B \to C, \mathrm{FHC}^{-1}(0) = B \to A \to D \to C \to D \to A$$



(a) **(b)**

Figure 3. (a) Labeling of $\mathrm{FHC}(0)$ **(b)** Labeling of $\mathrm{FHC}^{-1}(0)$

On $\mathrm{FHC}(1)$, coordinates arex, x', y, y', z, z' coordinates can be seen in Figure 4 and according to these coordinates fractals are placed around central part that is origin as seen in the Figure 5.

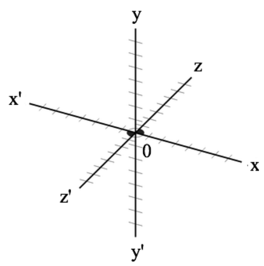


Figure 4. FHC coordinates

We call origin, y, z, x, y', z' and x' direction Part 0 (central part), Part I, Part II, Part III, Part IV, Part V and Part VI areas, respectively.

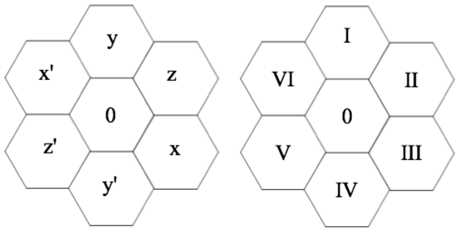
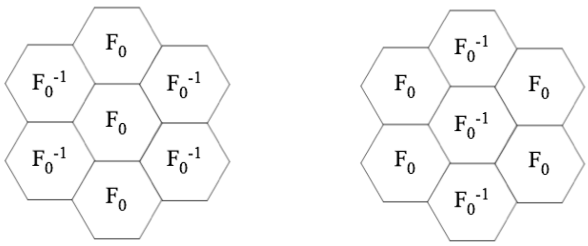


Figure 5. FHC coordinates and parts for placement of fractal

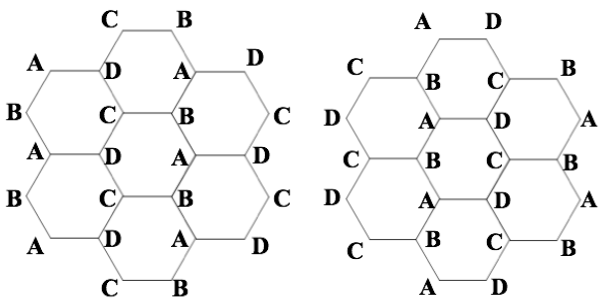
In Figure 6, how a placement of $\mathrm{FHC}(0)$ and $\mathrm{FHC}^{-1}(0)$ has done to construct $\mathrm{FHC}(1)$ and $\mathrm{FHC}^{-1}(1)$ can be seen.



(a) **(b)**

Figure 6. (a) F_1: main placement of FHC(1) **(b)** $F_1^{-1:}$ Inverse placement of FHC(1)

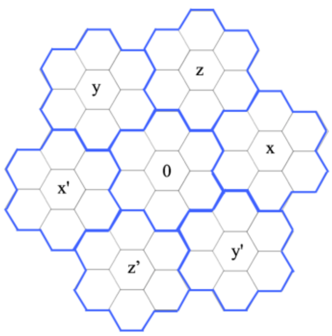
As shown in Figure 7, we find two different labeling of FHC for $n = 1$ using $\mathrm{FHC}(0)$ and $\mathrm{FHC}^{-1}(0)$.



(a) **(b)**

Figure 7. (a) Labeling of $\mathrm{FHC}(1)$ **(b)** Labeling of $\mathrm{FHC}^{-1}(1)$

On $\mathrm{FHC}(2)$, coordinates are x, x', y, y', z and z'coordinates placed around central part that is origin as seen in the Figure 8.



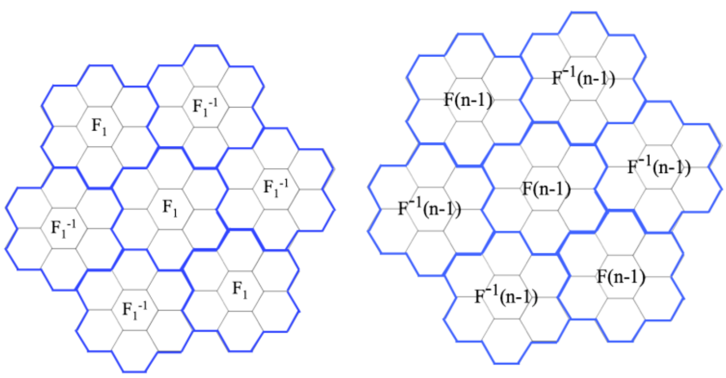


Figure 8. Coordinates of FHC(2). **Figure 9. (a)** Placement of FHC(1)s to construct FHC(2).
(b) Placement of FHC(n-1)s to construct FHC(n).

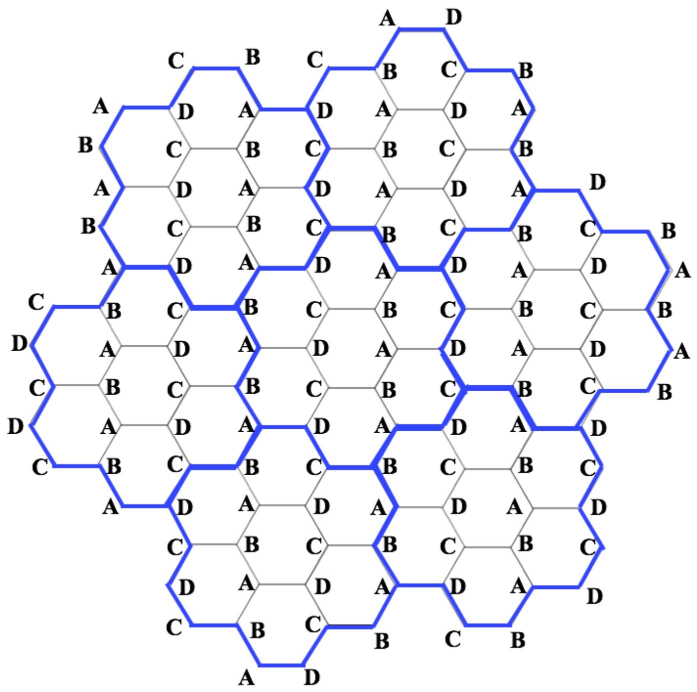


Figure 10. Labeling of FHC(2).

Coordinates of the fields change at one higher level. For F_1, coordinates are in the original position. For F_2, fields change one-step at counterclockwise, then for F_3, coordinates return back to its original position. For F_4, fields change one-step at clockwise, then for F_5, coordinates return back to its original position. This coordinate changes repeats every four level. It starts with original position, then moves one step counterclockwise, then returns original position and moves one step clockwise. Figure 5, Figure 8, Figure 11 and Figure 12 shows the placement of fractals according to given coordinates.

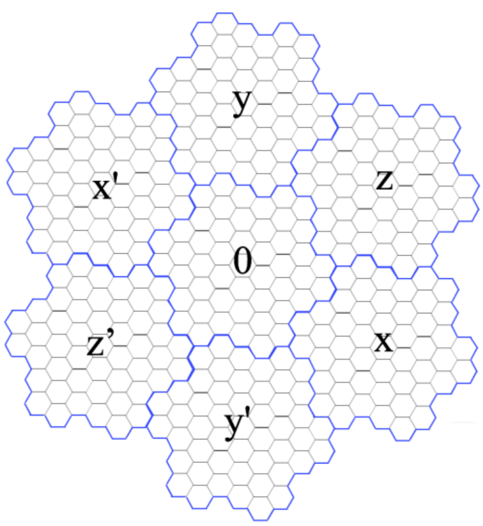


Figure 11. Coordinates of FHC(3).

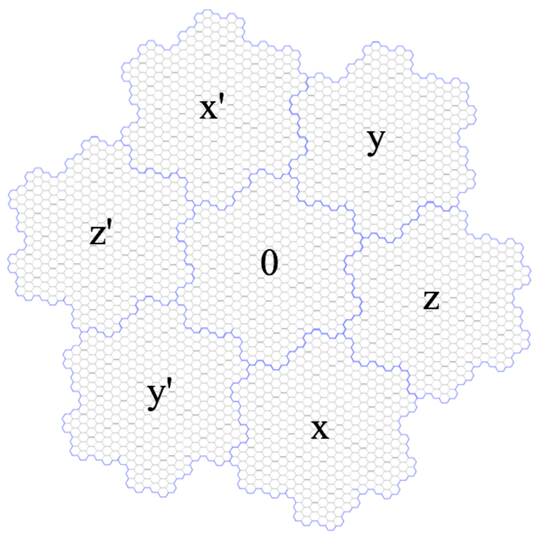


Figure 12. Coordinates of FHC(4).

Recursive Algorithm 1. (FHC) Firstly, we define $\mathrm{FHC}(0) = D \to C \to B \to A \to B \to C$, $\mathrm{FHC}^{-1}(0) = B \to A \to D \to C \to D \to A$.The following recursive algorithm is used for labeling the nodes of mesh structure FHC(n).

```
FHC(n, m)                        // if m=-1, then we calculate FHC^{-1}, if m=0, then we calculate FHC
    if level is 1
        if m!=-1                                         // construction of FHC(1)
            Copy FHC(0) to Part O, Part I and Part IV
            Copy FHC^-1(0) to Part II, Part III, Part V and Part VI
        else                                             // construction of FHC^{-1}(1)
            Copy FHC(0) to Part II, Part III, Part V and Part VI
            Copy FHC^-1(0) to Part O, Part I and Part IV
    else
        if m!=-1                                         // construction of FHC(n-1)
            Call FHC for Part O with level n-1
            Call FHC^-1 for Part II with level n-1
            Copy Part O to Part I and Part IV
            Copy Part II to Part III, Part V and Part VI
        else                                             // construction of FHC^{-1}(n-1)
            Call FHC for Part II with level n-1
            Call FHC^-1 for Part O with level n-1
            Copy Part II to Part III, Part V and Part VI
            Copy Part O to Part I and Part IV
end
```

The time complexity of the recursive algorithm 1 on FHC(n) can be written as $T(n) = 2T(n-1) + \theta(1)$. Solving this recurrence relation, obtained running time of is $T(n) = \theta(2^n)$.

3.2. FHCC Networks

In this subsection, we introduced the construction of FHCC(n).The Fractal Honeycomb network FHC(0) is a hexagon. As shown in Figure 13, we found two different labeling of FHCC for $n = 0$ choosing the starting node is A or C;

$$\text{FHCC}(0) = A \to B \to C \to D \to C \to B, \text{FHCC}^{-1}(0) = C \to D \to A \to B \to A \to D$$

On FHCC we used two different labeling for FHCC(0) and used these at FHCC(1).

(a) **(b)**

Figure 13. (a) Labeling of FHCC(0) **(b)** Labeling of $\text{FHCC}^{-1}(0)$

When we are using FHCC(0) while constructing FHCC(1), we are duplicating them to other parts of FHCC(1). If FHCC(0) is the Part I and $\text{FHCC}^{-1}(0)$ is the Part II for FHCC(1), duplicate Part I to Part III and duplicate Part II to Part IV is the first step for constructing FHCC.

The coordinates of FHCC is x, x', y and y' can be seen in Figure 14.

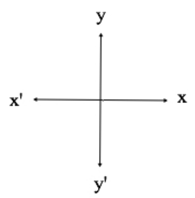


Figure 14. Coordinates of FHCC

The areas of FHCC according to coordinates as x, x', y and y' and as Part I, Part II, Part III and Part IV can be seen in Figure 15.

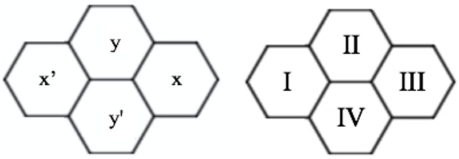


Figure 15. Areas of FHCC(1) **(a)** Coordinates **(b)** Parts.

Labeling FHCC(1) is the base for fractal HCC algorithm and how is the labeling done is shown in the Figure 16.

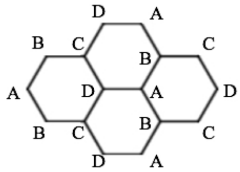
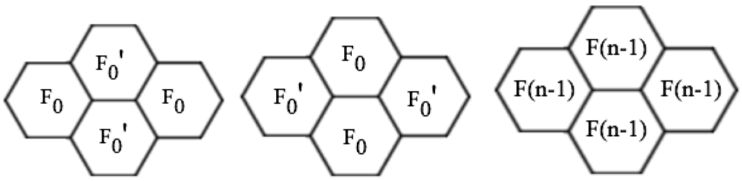


Figure 16. Labeling of FHCC(1) according to coordinates.

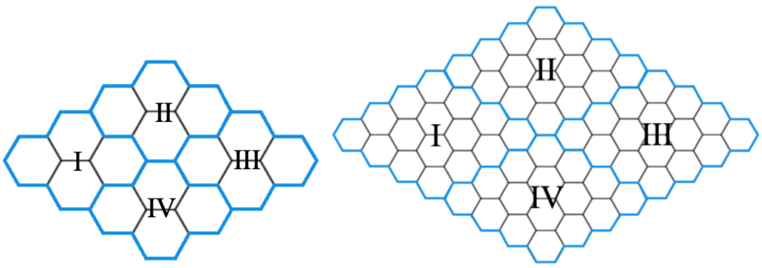
As shown in Figure 17 we could get two different labeling of $FHCC$ for $n = 1$ using $FHCC(0)$ and $FHCC^{-1}(0)$ according to given algorithms for FHCC.



(a) **(b)** **(c)**

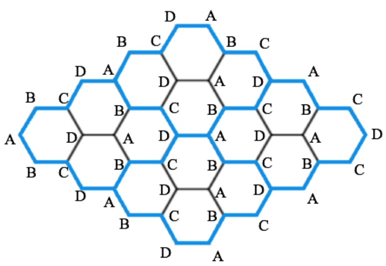
Figure 17. (a) F_1: main placement of FHCC(1) **(b)** F_1^{-1}: Inverse placement of FHCC(1) **(c)** F_n: main placement of FHCC(n)

Coordinates of FHCC have never been changed during the whole process. Parts for FHCC(2) and FHCC(3) are same with parts of FHCC(1). If Figure 18 is examined, it can be seen easily.



(a) **(b)**

Figure 18. Parts for level 2 and level 3 FHCC **(a)** FHCC(2) **(b)** FHCC(3).



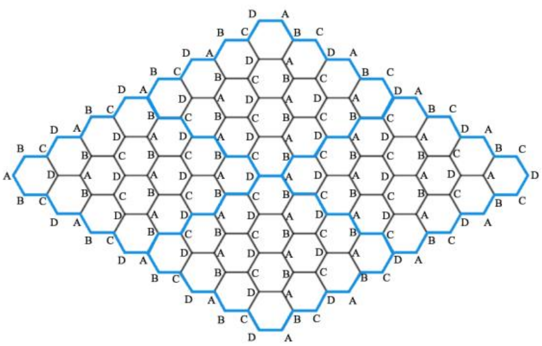


Figure 19. Labeling of FHCC(2). **Figure 20.** Labeling of FHCC(3).

Recursive Algorithm 2.(FHCC) Firstly, we define $\mathrm{FHCC}(0) = A \rightarrow B \rightarrow C \rightarrow D \rightarrow C \rightarrow B$, $\mathrm{FHCC}^{-1}(0) = C \rightarrow D \rightarrow A \rightarrow B \rightarrow A \rightarrow D$.The following recursive algorithm is used for labeling the nodes of mesh structure $\mathrm{FHCC(n)}$.

```
1. FHCC(n)
2.    if level is 1
3.          Copy FHCC(0) to Part I and Part III
4.          Copy FHCC^-1(0) to Part II and Part IV
5.    else
6.          Call FHCC for Part l with level n-1
7.          Copy Part l to Part ll-Part lll-Part IV
8. end
```

In the above algorithm, it can be used by exchanging third and fourth lines. As a result, different labelling (reverse labelling) can be achieved in $\mathrm{FHCC(n)}$.

Without loss of generality, the time complexity of the recursive algorithm 2 on FHCC(n) can be written as $T(n) = T(n-1) + \theta(1)$ (if initial level is 2). Solving this recurrence relations, obtained running time of is $T(n) = \theta(n)$.

4. Conclusion

We get perfect matching of labeling of nodes in Fractal Honeycomb meshes (FHC and FHCC) for dimension one and two using 2-bit gray code. Therefore, if desired to construct fractal meshes, it would be better to use recursive algorithm 2. Because, recursive algorithm 1 has exponential running time.

Acknowledgement. The authors are very grateful to Professor Ali Karcı (The University of İnönü, Turkey) for his valuable suggestions, which helped to improve the paper significantly.

References:

Hales T. C. (2001) The Honeycomb Conjecture. *Discrete and Computational Geometry* 25(1):1-22.

Nocetti F. G., Stojmenovic I., Zhang J. (2002) Addressing and Routing in Hexagonal Networks with Applications for Tracking Mobile Users and Connection Rerouting in Cellular Networks. *IEEE Trans. Parallel Distrib. Syst.* 13(9): 963-971.

Carle J., Myoupo J. F., Seme D. (1999) All-to-all Broadcasting Algorithms on Honeycomb Networks and Applications.*Parallel Process. Lett.*9(4): 539-550.

Manuel P., Rajan B., Rajasingh I., MC. M. (2008)On Minimum Metric Dimension of Honeycomb Networks.*J. Discret. Algorithms* 6(1): 20-27.

Lester L. N., Sandor J. (1985) Computer Graphics on a Hexagonal Grid.*Comput. Graph*8(4): 401-409.

Rajan B., William A., Grigorious C., Stephen S. (2012) On Certain Topological Indices of Silicate , Honeycomb and Hexagonal Networks.*J. Comp. Math. Sci*3(5): 530-535.

Boudjemai A., Amri R., Mankour A., Salem H., Bouanane M. H., Boutchicha D. (2012) Modal Analysis and Testing of Hexagonal Honeycomb Plates Used for Satellite Structural Design. *Mater. Des.*35: 266–275.

Engelmary G. C., Cheng M., Bettinger C. J., Borenstein J. T., Langer R., Freed L. E. (2008) Accordion-like Honeycombs for Tissue Engineering of Cardiac Anisotropy. *Nature Materials* 7: 1003-1010.

Mandelbrot B. (2004)Fractals and Chaos : The Mandelbrot Set and Beyond, Springer Science & Business Media, United States of America.

Kluge T. (2000) Fractals in Nature and Applications. https://kluge.in-chemnitz.de/documents/fractal/node2.html.Accessed 13 December 2018.

Joye Y.(2005) Evolutionary and Cognitive Motivations for Fractal Art in Art and Design Education. *International Journal of Art and Design Education*2(2): 175-185.

Keller J., Chen S. (1989) Texture Description and Segmentation Fractal Geometry. *Computer Vison, Graphics, and Image Proccessing*45: 150–166.

Karci A., Selçuk B. (2014) A new hypercube variant : Fractal Cubic Network Graph.*Engineering Science and Technology, An International Journal*18(1): 32-41.

Meier J., Reiter C. A. (1996) Fractal Representations of Cayley Graphs.*Caos & Graphics.* 20(1): 163-170.

Warchalowski W., Krawczyk M. J. (2017) Line Graphs for Fractals.*Commun. Nonlinear Sci. Numer. Simul.* 44: 506-512.

Fiala J., Hubička J., Long Y., Nešetřil J. (2017) Fractal Property of the Graph Homomorphism Order.*Eur. J. Comb.* 66: 101–109.

Jaggard D. L., Bedrosian S. D. (1987) A Fractal-Graph Approach to Large Networks. *Proceeding of IEEE* 75(7): 966–968.

Montiel M. E., Aguado A. S., Zaluska E. (1995) Topology in Fractals.*Caos, Solitons & Fractals* 7(8): 1187-1201

Brown J. I., Hickman C. A., Nowakowski R. J. (2003)The Independence Fractal of a Graph. *J. Comb. Theory*87: 209-230.

Ejov V., Filar J. A., Lucas S. K., Zograf P. (2007) Clustering of Spectra and Fractals of Regular Graphs.*J. Math. Anal. Appl.*333: 236-246.

Science N., Phenomena C., Komjáthy J., Simon K. (2011) Generating Hierarchial Scale-Free Graphs from Fractals.*Chaos , Solitons Fractals* 44: 651–666, 2011.

Cho, H. J., & Hsu, L. Y. (2003). Generalized honeycomb torus. Information Processing Letters, 86(4), 185-190.

Yang, X., Evans, D. J., Lai, H., & Megson, G. M. (2004). Generalized honeycomb torus is Hamiltonian. Information Processing Letters, 92(1), 31-37.