Shortest Distance between Two Vertices in a Honeycomb Graph

Suman Kumar Chanda[1*]

1* Department of Computer Science, Panchakot Mahavidyalaya, Purulia, e-mail: sumanchanda1991@gmail.com

Introduction

Honeycomb structures are natural or man-made structures that have the geometry of a honeycomb to allow the minimization of the amount of used material to reach minimal weight and minimal material cost. The geometry of honeycomb structures can vary widely but the common feature of all such structures is an array of hollow cells formed between thin vertical walls. The cells are often columnar and hexagonal in shape **[4]**.

Man-made honeycomb structural materials are commonly made by layering a honeycomb material between two thin layers that provide strength in tension. This forms a plate-like assembly. Honeycomb materials are widely used where flat or slightly curved surfaces are needed and their high strength-to-weight ratio is valuable. They are widely used in the aerospace industry for this reason, and honeycomb materials in aluminum, advanced composite materials have been featured in aircraft and rockets since the 1950s. They can also be found in many other fields, from packaging materials in the form of paper-based honeycomb cardboard, to sporting goods like skis and snowboards.

Natural honeycomb structures include beehives, honeycomb weathering in rocks, tripe, and bone. Man-made honeycomb structures include sandwich structured composites with honeycomb cores. Man-made honeycomb structures are manufactured by using a variety of different materials, depending on the intended application and required characteristics, from paper or thermoplastics, used for low strength and stiffness for low load applications, to high strength and stiffness for high performance applications, from aluminum or fiber reinforced plastics. The strength of laminated or sandwich panels depends on the size of the panel, facing material used and the number or density of the honeycomb cells within it. Honeycomb composites are used widely in many industries, from aerospace industries, automotive and furniture to packaging and logistics. The material takes its name from its visual resemblance to a bee's honeycomb – a hexagonal sheet structure.

Objective

In this work we proposed an algorithm to find the minimum distance between any two vertices of two dimensional hierarchical honeycomb structures. There are many paths between two vertices of a honeycomb graph; here we calculate the minimum path distance of them.

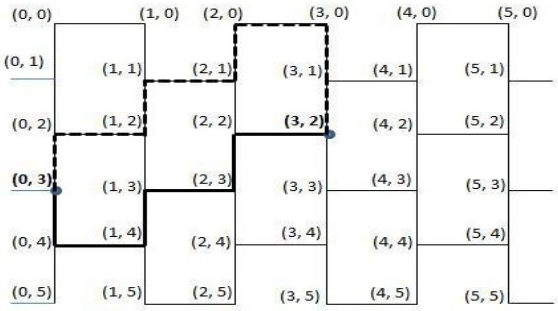

Fig.1: Different paths between two vertices (0, 3) and (3, 2)

- ----- Distance 8
- —— Distance 6 (minimum)

In Fig.1 we consider the two vertices (0, 3) and (3, 2). We can see that there are many paths between them with distance 6, 8 etc. In this work we consider the minimum distance i.e. 6 (Bold Line) and calculate it.

Definition

A graph G is defined as an ordered pair *G= (V, E)*, where V= {v_1,….,v_n}is called the graph's *vertex- set* and$E = \{e_1, \ldots, e_m\} \subset \{\{x, y\} | x, y \in V\}$ is called the graph's *edge-set***[1].**

We say that a graph G_1 is an *induced sub graph* of a graph *G* if G_1 is isomorphic to a graph whose vertex set V_1 is a subset of the vertex set *V* of *G*, and whose edge set E_1 consists of *all* the edges of *G* with both end vertices in V_1. Note that by our definition a graph is always an induced sub graph of itself. It is important to note that not all sub graphs of a graph are induced sub graphs. We say that the graph G_1 is the *graph induced by the vertices V_1 in G**[2]**.*

A honeycomb graph is a finite induced sub graph of the infinite graph associated with the two- dimensional hexagonal grid **[3].**

Two dimensional prismatic cellular materials of periodic microstructures are called honeycombs. The hexagonal structure is the most commonly used honeycomb geometry.

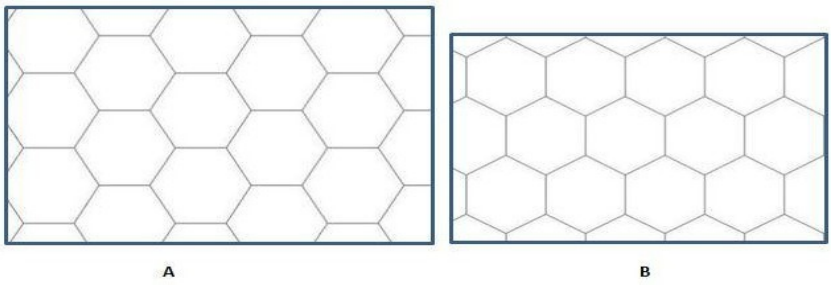

Fig.2: Different Honeycomb Graphs

Co-ordinate system of proposed honeycomb graph

The axes of honeycomb cells are always quasi-horizontal, and the non-angled rows of honeycomb cells are always horizontally (not vertically) aligned.

The x-axis here is an easy one to assign. We can start from any vertex and go straight to the right, incrementing the value of i along the way and go straight to the bottom, incrementing the value of j.

Neighbor of a Vertex

Let (i,j) be any vertex of an infinite Honeycomb graph .As it is a regular G_3 graph , each vertex has three neighbors. If there is a direct edge between (i,j) and (i+1,j) then neighboring vertex are(i,j- 1),(i,j+1) and (i+1,j) else the neighboring vertex are (i,j-1),(i,j+1) and (i-1,j).

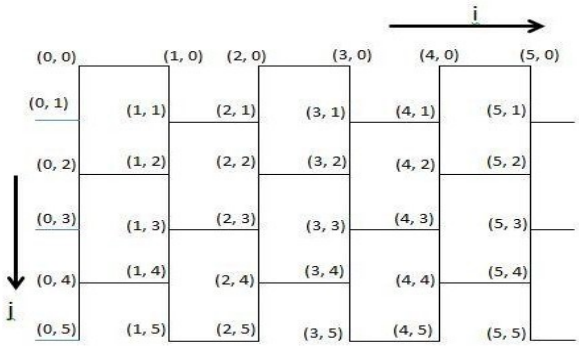

Fig.3: Co-ordinate representation of Honeycomb Graph

Properties of honeycomb networks

Honeycomb networks can be built from hexagons in various ways. The honeycomb network HC(1)is a hexagon. The honeycomb network HC(2)is obtained by adding six hexagons to the boundary edges of HC(1). Inductively, honeycomb network HC(n) is obtained from HC(n−1) by adding a layer of hexagons around the boundary of

HC(n−1). For instance, Fig. 1(c) is HC(2). The parameter n of HC(n)is determined as the number of hexagons between the centre and boundary of HC(n). The number of vertices and edges of HC (n) are 6 and 9 −3n respectively.

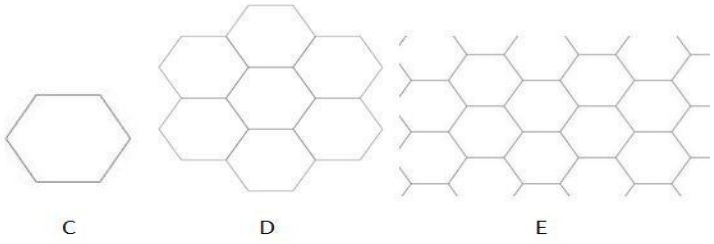

Fig. 4: Formation of Honeycomb Grid

- C - HC(1)
- D - HC(2)
- E - HC(n)

This graph is sometimes referred as G3 as the degree of each vertex is 3. Hexagonal cell honeycombs are known to be bending dominated structures, which is good for flexible structural design. Hexagonal cell structures are known to be flexible in both axial and shear directions. Moreover, hexagonal honeycombs can easily be tailored to have targeted in-plane properties by changing cell angles.

Different isomorphic representation of honeycomb graph

Let G={V,E} and G'={V', E'} be graphs. G and G′ are said to be isomorphic if there exist a pair of functions V: V →V' and g: E → E' such that f associates each element in V with exactly one element in V′ and vice versa; g associates each element in E with exactly one element in E′ and vice versa, and for each v ∈ V, and each e ∈ E , if v is an endpoint of the edge e, then f(v) is an endpoint of the edge g(e).

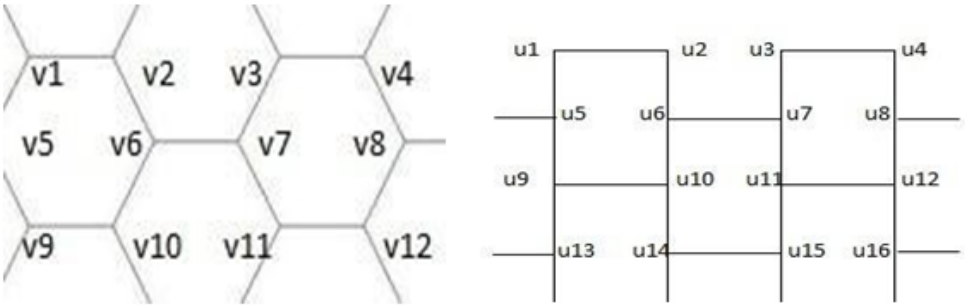

Fig. 5: Different Isomorphic representation of Honeycomb Graph

Working principle of proposed algorithm

Let the co-ordinates of the two vertices A and B of an infinite honeycomb grid be ( ) and ( ) respectively. In our proposed algorithm we calculate the shortest distance between these two vertices A and B. The working principle of the proposed algorithm is as follows:

Case-I. $(i_1 = i_2)$
$Distance = | j_1 - j_2 |$.

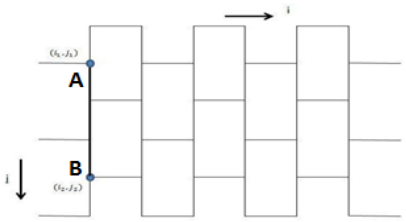

Fig.6: Vertices *A* and *B* satisfying Case-I

Case-II. () and ( )

Then we check if there is any direct path between ( ) and ( ). If it is true then return distance d=2*|()|-1, else return d=2*|()|+1.

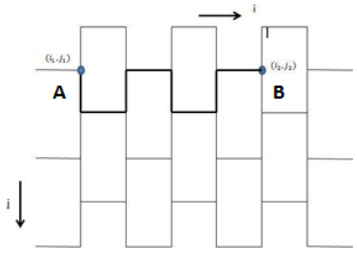

Fig.7: Vertices A and B satisfying Case-II

Case-III. $(i_1 \neq i_2)$ and $(j_1 \neq j_2)$

Then check whether these two given nodes either in stair case or not by checking this if $|(i_1-i_2)| == |(j_1-j_2)|$. If true then these two points are in stair case and distance will be $2*|(i_1-i_2)|$.

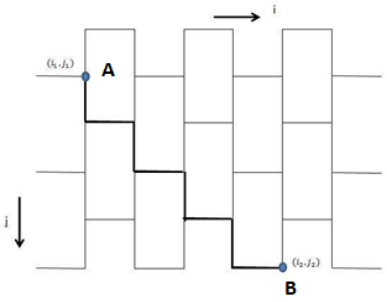

Fig.8: Vertices A and B satisfying Case-III

Case-IV. If these two points are not in stair case then we have to find the nearest stair points (i j) to (i2, j2). From this point we can calculate vertical or horizontal distance between (i, j) and (i2,j2). Add two distances.

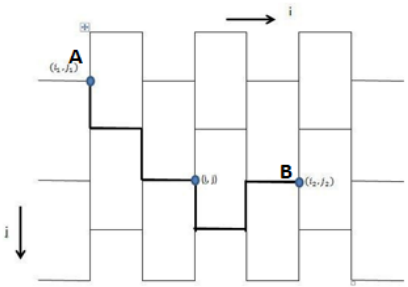

Fig.9.Vertices A and B satisfying Case-IV

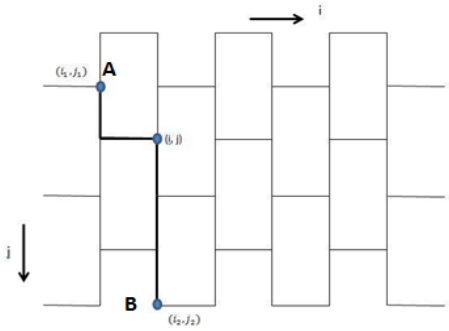

Fig.10. Vertices A and B satisfying Case-V

Algorithm

Input: (i_1,j_1) and (i_2,j_2),where (i_1,j_1) and (i_2,j_2) are co-ordinates of two vertices of an infinite graph.

Output: Shortest Distance between two given vertices.

Pseudo Code:

Step 1.
If$(i_1==i_2 \&\& j_1==j_2)$
then distance will be *0.*
*Step2.*If$(i_1==i_2 \&\& j_1!=j_2)$
then distance will be$d=|(j_1-j_2)|$, *where d is the shortest distance.*
Step 3.If$(i_1 !=i_2 \&\& j_1==j_2)$
then distance can be measured as
$d=distance(i_1,i_2,j_1)$,*d is the calculated shortest distance.*
Step 4.If$(i_1!=i_2 \&\& j_1!=j_2)$

then check whether if(|(i1-i2)|==|(j1-j2)|),if itis true then distance can be calculated as

d=|(i1-i2)|+|(j1-j2)|.

Otherwise check if (|(i1-i2)<|(j1-j2)|)

then distance can be calculated as

set that value of new co-ordinate positionas i=i2.

if(j1<j2).

{

} j=j1+|(i1-i2)|.

else

{

}

j=j1-|(i1-i2)|.

when (|(i1-i2)|>|(j1-j2)|) then

set the value of another new position as j=j2.

i=i1+|(j1-j2)|.

}

and d1=|(i1-i)|+|(j1-j)|.

check if(i2==i)

d0=|(j

-j2)|; else

if(j==j2)

d0=distance(i,i2,j2);

d=d0+d1;

hered1 and d2 are two intermediate results & d is the final calculated shortest distance.

Distance(i1,i2,j1)

Step1 :Initialize d=0 and m=0

Step 2: If |i1-i2| is even then distance d= d=2*|(i1-i2)|

Step 3: Else m=check(i1,j1)

Step 4: if m=1 then return d= d=2*|(i1-i2)|+1.

Step 5: else return d= d=2*|(i1-i2)|-1.

Check(i1,j1)/* Checks whether there exist any edge between i1 and j1 */

Step1 :Initialize v=0.

Step 2:if (i1+1) is odd and j1 is even o (i1+1) is even and j1 is odd then

Step 3: set v=1.

Step 4: return v.

Limitation of this Algorithm

Although this algorithm can compute the distance between any two vertices, but the graph must obeys the proposed co-ordinate system. See the following figure for demonstration**.**

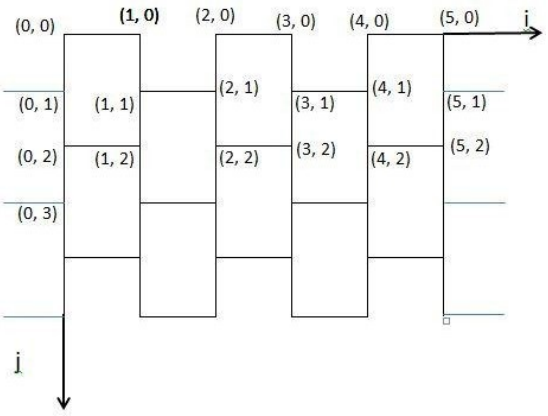

Fig.11 Proposed Co-ordinate System

References

We have used many resources and for that we are grateful to all the people concerned.

Given below are the names of some books, which we have used development and documentation of the project.

i). Kinkar Ch. Das: History and Application of Spectral Graph Theory,
ii). Department of Mathematics, Sungkyunkwan University
iii). http://www.wolframalpha.com/input/?i=vertex-induced+subgraph
iv). http://mathworld.wolfram.com/Honeycomb.html
v). http://en.wikipedia.org/wiki/Honeycomb