Checking self-duality

Mudit Rathore[1], Ishani Mishra[2], and Gugloth Lavanya[3]

[1]Indian Institute of Information Technology Allahabad, ICM2015502
[2]Indian Institute of Information Technology Allahabad, IWM2015008
[3]Indian Institute of Information Technology Allahabad, ISM2015003

***Abstract*— Duality of planar graphs has applications in graph theory as well as several other areas of mathematical and computational study, in geographic information systems, flow networks, computer vision and many more. Flow networks (such as the networks showing how water flows in a system of streams and rivers) are dual to cellular networks describing drainage divides. In computer vision, digital images are partitioned into small square pixels, each of which has its own color. The dual graph of this subdivision into squares has a vertex per pixel and an edge between pairs of pixels that share an edge; it is useful for applications including clustering of pixels into connected regions of similar colors.**
This paper deals with an algorithm to check self-duality of a graph given its adjacency matrix representation.

I. INTRODUCTION

A Graph [1] G is defined as an ordered pair of a finite set V of vertices and a finite set E of edges, where a vertex is a node and an edge is a connection between any two vertices. A graph can be represented using Adjacency Matrix and Incidence Matrix.

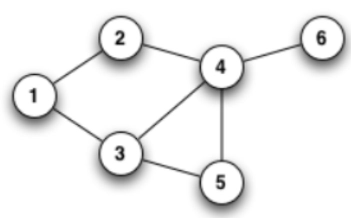

Fig. 1. A simple undirected graph

When a connected graph can be drawn without any edges crossing, it is called planar [2]. When a planar graph [3] is drawn in this way, it divides the plane into regions called *faces*. Hence faces of a graph may be defined as the regions within the graph, divided by the edges of a planar graph. In the graph shown below (Fig. 2.), there are three faces:

1) 2-5-4-3-2

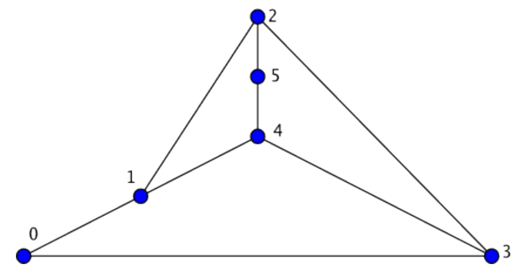

Fig. 2.

2) 2-5-4-1-2
3) 0-1-4-3-0

The graph shown below does not look planar because edges at the diagonals seem to be crossing each other.

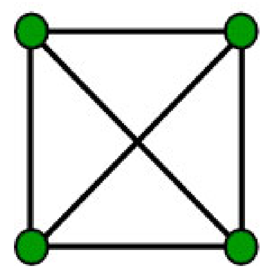

Fig. 3. A graph

But the graph is planar, just as we can see below:

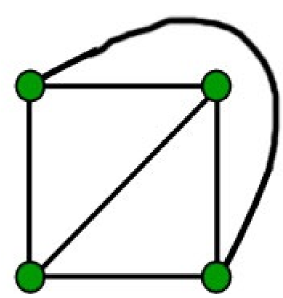

Fig. 4. Planar Graph

Since the point of intersection of the diagonals is not a vertex, the edge can be stretched out of the 4 edges bounds of the square and can be drawn from outside. Note that the edge shifted still connects the same pair of vertices.

Faces can now be computed easily. If we name the four vertices as A B C and D, taken clockwise from top left, the faces are:

- A B D A
- A B C A
- B C D A
- A B C D A

Euler's Formula for planar graphs
For any connected planar graph G = (V, E), the following formula holds [4]:

$$V + F - E = 2$$

where F stands for the number of faces.

For any disconnected planar graph G = (V, E) with k components, the following formula holds:

$$V + F - E = 1 + k$$

We know that edges of a planar graph divides the region bounded inside the graph into faces. Now, if we see the faces as vertices and the edges common to faces as edges, and we create a graph in this way, the graph obtained is known as *dual graph*. It is represented as **G***.

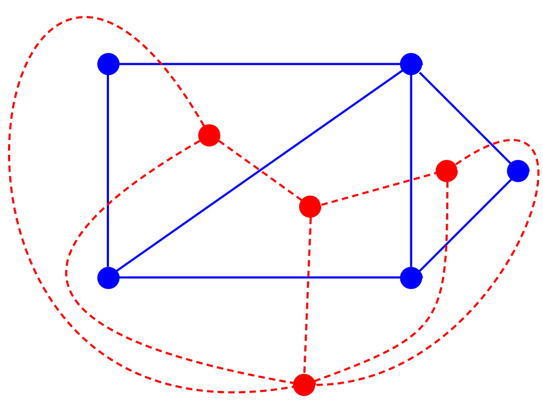

Fig. 5. Dual Graph

In the above figure, the graph with blue edges is the original planar graph. Each face is marked by a red vertex in the corresponding dual graph. All these vertices have been connected by dotted red lines. One edge of the dual graph corresponds to one edge of the original graph separating the faces in the original graph.

It must be kept in mind that the dual graph should also be a planar graph meaning that there should be no edges crossing in it. The figure depicted above

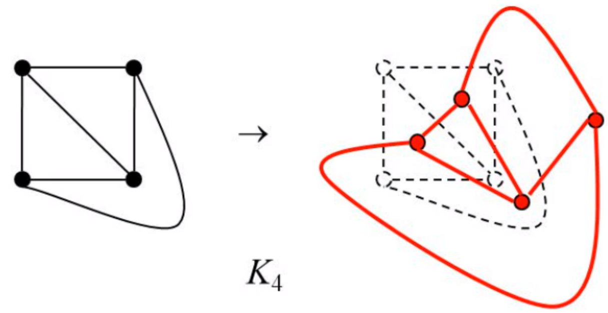

Fig. 6. Dual Graph of a planar graph with 4 vertices

shows a graph with 4 vertices (left) and its dual graph (right).

A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity. Such graphs are called isomorphic graphs [5]. Two graphs which contain the same number of graph vertices connected in the same way are said to be *isomorphic*.

A plane graph is said to be *self-dual* if it is isomorphic to its dual graph. A graph that is dual to itself is a self-dual graph [6]. Wheel graphs are self-dual, as are the examples illustrated below. Naturally, the skeleton of a self-dual polyhedron is a self-dual graph. Since the skeleton of a pyramid is a wheel graph, it follows that pyramids are also self-dual.

For example, shown below is a graph with 6 faces. When the dual of the above graph is made, it looks

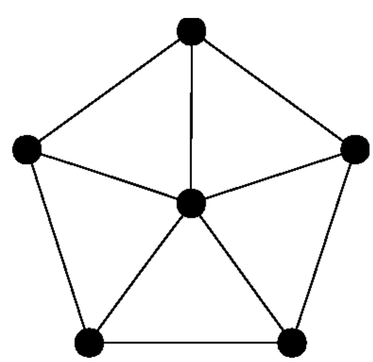

Fig. 7. A pentagon planar graph with 6 vertices

something like the figure shown below. The figure

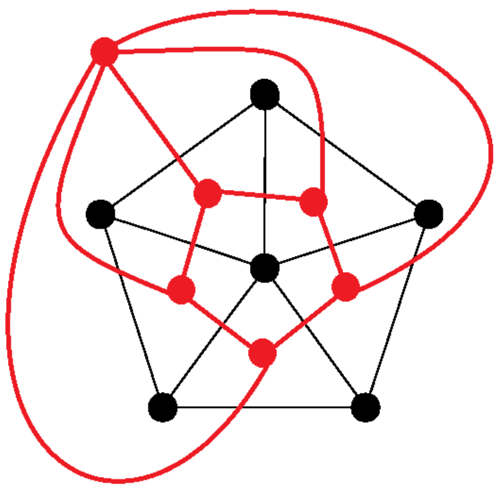

Fig. 8. Dual graph of Fig. 7. drawn on the original graph

above can be rearranged to form a graph shown below. The vertex symbolizing the external face can be shifted inside the pentagon and can be depicted as shown below. Now we can see that figure 7 and figure 9 are isomorphic and hence self dual.

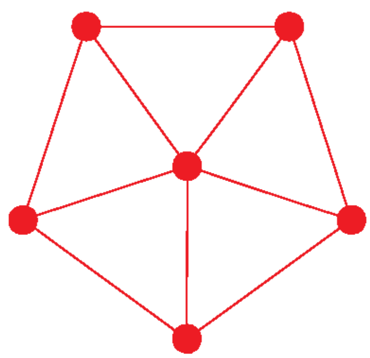

Fig. 9. Dual graph of Fig. 7.

II. METHODOLOGY

This paper deals with checking if the given graph is self-dual or not.

A. Input

The user is made to input the number of the vertices of the graph and the adjacency matrix of the graph.

B. Proposed Algorithm

For the algorithm that we propose here, we first have to find the number of faces and the edges that enclose a face. Then we need to generate the dual graph of the given graph which is formed by considering faces as the vertices of the graph and then considering every edge as the faces then input an edge int he graph thus formed with the faces as the vertices of the graph between the faces with shares the edges in common as the enclosing boundary. After having successfully generated the dual of the given input graph, we check whether the graph obtained is isomorphic to the input graph. The graph given as input and its obtained dual graph must be isomorphic for the self-duality to hold, otherwise the graph is not self dual. The proposed algorithm for checking the self-duality is as follows:

The first algorithm finds all possible circuits from all vertices one by one, finally considering non redundant minimal circuits or cycles from all vertices.

1) Select a vertex.
2) Find all cycles from the vertex. Once all cycles from a vertex have been found out, remove the vertex. So now cycles are found from other vertices will not include the previously removed vertices.
3) Keep storing cycle in a vector.
4) Repeat step 2 - 3 for all vertices.
5) Sort the cycles in accordance to cycle length.
6) Consider the shortest cycle.
7) Add it to face vector if it hasn't appeared before in the vector.
8) Consider the next cycle from the sorted list of cycles.
9) Repeat steps 7 – 8 till all cycles have been considered.
10) At the end, non redundant unique cycles will be in the face array and Those will be all the faces.
11) The overall face can be added which is actually the longest cycle containing all vertices.

- *Finding all possibles cycles*

The algorithm to find all possible cycles from all vertices. It is a recursive approach to identify all possible cycles present in the graph from every vertex to every other vertex. A vector declared globally stores all cycles.

1) Choose a vertex.
2) Choose adjacent vertices of a vertex.
3) Find the path between the vertex and above chosen adjacent vertex.
4) Store it in cycle vector along with the start vertex at the end of the above found path. This completes the cycle and stores it.

We repeat the same above approach for all vertices. In order to reach from start vertex to the destination vertex, we change the start vertex to the next reachable vertex in every function call while keeping the destination fixed in order to trace the path. Towards the end of the algorithm, when all cycles have been calculated and stored. Further computing the minimal unique cycle can be done as stated in the algorithm above.

Till Now we have Identified the vertices of the dual graph now we have to identify the edges of the dual graph.

1) Take an edge from the Edge Set.
2) Check for the faces those have chosen edge in common
3) Connect the faces having the chosen edge as common edge.
4) Remove the above chosen edge from the edge set.
5) Repeat these steps until the edge set becomes empty

Now we have the Input graph and the dual graph generated with the help of the faces.if we prove that these graphs are isomorphic then we will be able to prove the self duality of the graph. The Algorithm for checking of the self duality

1) Input the Two graph
2) check_isomorphism —— this function will check whether the two graphs are isomorphic
   a) if the graph size are equal check whether the given permutation of graph 1 is equal to the permutaion of graph 2 or not.
   b) if they are equal then they are isomorphic if for no permutation they are equal they are not isomorphic.
3) If they are not isomorphic they are not self dual and vice versa.

III. RESULTS

The objective of this paper was to identify if the given graph in form of adjacency matrix is a self dual graph or not. The Proposed algorithm will give an output of "Is Self Dual" for the graph that has isomorphic dual graph and the input graph and "Not Self Dual" for the graph that does not have isomorphic dual graph and input graph.

Time complexity of the above for the generation of the faces algorithm is $O(V^V)$ because the maximum adjacency of a vertex can be V - 1. Thus finding cycles for each vertex, comparing all cycles with previously occurred faces involve V x V x V .... (V-1) times computations, hence the time complexity is $O(V^V)$.

If the number of faces is F, then Algorithm for generation of the dual graph takes $O(F^2)$ as it considers each pair of faces to decide whether an edge is to be added or not. The space complexity is dominated by the number of elements in the adjacency matrix, $O(V^2)$.

IV. DISCUSSION

Because the dual graph depends on a particular embedding, the dual graph of a planar graph is not unique, in the sense that the same planar graph can have non-isomorphic dual graphs. In the picture depicted below, the blue graphs are isomorphic but their dual red graphs are not. The upper red dual has a vertex with degree 6 (corresponding to the outer face of the blue graph) while in the lower red graph all degrees are less than 6.

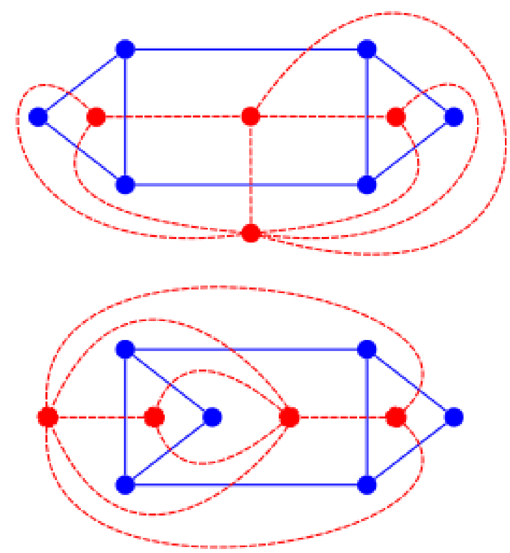

Fig. 10.

REFERENCES

[1] J. Cook, M. Wootters, V. Williams, R. Verma, S. Hildick-Smith, and G. Valiant, *Graphs.* [On-line]. Available: https://web.stanford.edu/class/cs161/Lectures/Lecture9/CS161 Lecture09.pdf. [2017]

[2] V. Adamchik *Graph Theory* [On-line]. Available: http://www.cs.cmu.edu/adamchik/21-127/lectures/graphs.pdf [2005]

[3] R. J. Trudeau. *Introduction to graph theory*. Courier Corporation [2013].

[4] *Planar Graphs and Euler's Formula* [On-line]. Available: https://www.math.upenn.edu/ mlazar/math170/notes05-3.pdf [2016]

[5] Weisstein, Eric W. *Isomorphic Graphs*. (2002)

[6] Weisstein, Eric W. *Self-Dual Graph*. (2000).