

IMPLEMENTATION OF GREEDY ROUTING ALGORITHM FOR HAMILTONIAN CYCLE FROM QUASI SPANNING TREE OF FACES

P. RAMANA VIJAYA KUMAR¹ AND BHUVANA VIJAYA

ABSTRACT. In this paper the implementation of greedy routing algorithm is proposed for spanning of tree for faces. Here this algorithm will provide routing for Hamiltonian Cycle from Quasi Spanning Tree of Faces. In this Hamiltonian cycle two lemma theorems are introduced. These lemma theorems will describe only the expansion of tree for faces. Hence the routing procedure is introduced in these theorems for better result. Generally, there are 4 to 6 sides of faces in the graph that are hold conjunctively and this is introduced by Goodey. In the same way, the 3 connected cubic planar graphs will show that if 2 coloured faces are used then the vertex is incident to the two red faces and one blue face. Hence all the red coloured faces will have 4 to 6 sides and in the same way all blue coloured faces will have 3 to 5 sides. The proposed routing algorithm will reduce the contracting of each colour based on vertex and in the same way proper quasis panning tree of faces. In proposed algorithm, the parity of spanning tree will decided the arbitrary face based on even degree. Hence the greedy routing algorithm for Hamiltonian cycle from quasi spanning tree of faces will produce effective output compared to lemma 1 and lemma 2 theorems.

¹*corresponding author*

2010 *Mathematics Subject Classification.* 05C05.

Key words and phrases. Greedy routing algorithm, Hamiltonian cycle, uniquely hamiltonian, uniquely traceable, Bondy-Jackson conjecture, cubic graph, girth, exhaustive generation.

1. INTRODUCTION

Greedy routing algorithm consists of finite integral multipliers. Here sub graphs are used in the maximal tree and this sub graph is represented as G [1]. Here the both trees are distributed with each other, if the edges are labelled suitably. Here the distinction of maximum number of trees is based on the vertices and this is introduced by the Cayley. Generally, a graph G is constructed based on the vertices of spanning tree. By using single edge G the spanning tree is evaluated. Here diagonal matrix is introduced to define the vertex of system. The spanning tree determines the difference between the both incidence matrix and adjacency matrix [2]. Hence all the vertices are included in the sub graph G . For any single tree graph the diameter D is represented as

$$\text{diam}(T(G)) \leq \min\{n - 1, m - n + 1\}.$$

The diameter of spanning graph depends on the two trees either T_1 or T_2 and it is represented as:

$$d(T_1, T_2) = n - 1 - |E(T_1) \cap E(T_2)| = |E(T_1) \Delta E(T_2)|/2.$$

The operation of tree for graph is represented as

$$T : G \rightarrow G,$$

and the resultant matrix is the combination of rows and columns. This will determine the value of sub graph and vertex in effective way. The product value is obtained after determining the spanning tree calculation. The product value calculation is introduced by Cauchy-Binet. All this calculation is based on the calculation of both vertices and adjacency. Here the cycle that is used to formulate the process is very natural. An intersection form appears by using set of cycles. This will count the edges of cycle in the sign. Here, the edge is connected to the spanning of tree to identify the sub graphs and determine the paths from sub graphs.

Here, the value of sub graph is not equal to the geometric cycle. The spanning tree graph will use the iterated tree graph sequence which is given as $G, T(G)$ [3]. Basically, a graph is planar which has no edge crossings. In plane graph, a face is connected to a region with three edges. By keeping vertex for every face a graph is constructed generally. Here multiple graphs are connected to regions in a particular sequence. In multi graph, the trees has no disconnecting

edges and each edge will have a cycle to perform its operation. Therefore, the spanning tree counting procedure is performed based on the number of cycles. Here the value of determinant of a cycle is equal to the incidence matrix. Based on the integer, the algebraic cycles perform its operations.

2. HAMILTONIAN CYCLE FROM QUASI SPANNING TREE OF FACES

Let G be a graph in the class of 3-connected cubic planar graphs that has a set C of faces such that every vertex in G is incident to one face in C and to two faces not in C . We refer to the faces in C as blue faces and to the faces not in C as red faces. Let H be the corresponding reduced graph obtained by contacting the faces in C to single vertices. A spanning tree of faces in H is a set D of faces of H such that no two faces in D share an edge, and such that if the T be the graph with vertices corresponding to vertices in H and faces in D , and edges joining the vertices corresponding to faces d in D to the vertices in H incident to d , then T is a tree [4].

A quasi spanning tree of faces in H is a set D of faces of H and a set V of vertices in H such that no two faces in D share an edge, every vertex of H not in V has even degree, say $2r$, and is surrounded by r faces in D , and such that if we let T be the graph with vertices corresponding to vertices in V and faces in D , and edges joining the vertices corresponding to faces d in D to the vertices in V incident to d , then T is a tree. The vertices of H not in V are called quasi vertices, and a proper quasi vertex is a quasi vertex of degree 4 such that none of the 4 faces surrounding it is a digon (i.e., has only two sides). A proper quasi spanning tree of faces is a quasi spanning tree of faces such that all of its quasi vertices are proper quasi vertices.

Given a spanning tree of faces in H , it may assume the external face is not in D , and traverse the perimeter of the spanning tree of faces, to obtain a Hamiltonian cycle in G that has all faces of the collapsed set C inside. Given a quasi spanning tree of faces in H , it may assume the external face is not in D , and traverse the perimeter of the quasi spanning tree of faces, to obtain a Hamiltonian cycle in G such that the faces of the collapsed C are inside the cycle for vertices in V and outside the cycle for vertices not in V (quasi vertices) [5]. This gives the following.

Proposition 2.1. *The reduced graph H has a spanning tree of faces with the external face not in D if and only if G has a Hamiltonian cycle with the external red face outside, with all blue faces inside and such that no two red faces sharing an edge are both inside.*

Proof. The reduced graph H has a quasi spanning tree of faces with the external face not in D if and only if G has a Hamiltonian cycle with the external red face outside, with all blue faces corresponding to vertices in V inside, with all blue faces corresponding to vertices not in V (quasi vertices) outside, and such that no two red faces sharing an edge are both inside. \square

Lemma 2.1. *If T has at least two vertices inside and satisfies the invariant property, then it is possible to select a triangle T that is a face inside of T and collapse T to a single vertex in such a way that T still satisfies the invariant property.*

Proof. Suppose triangle T_1 inside of T contains at least two vertices inside, and there is no triangle inside of T_1 that is not a face. Writing $T_1 = v_1v_2v_3$, it claim that v_1 has at least two distinct neighbours v_4 and v_5 inside of T_1 . Otherwise, if v_1 has no such neighbours, then v_1 belongs to a triangle inside of T_1 that has an edge v_2v_3 parallel to the side of T_1 , contrary to the assumption that there is no digon inside of T that is not a face; and if v_1 has only one such neighbour v_4 inside of T_1 , then $v_2v_3v_4$ is a triangle inside of T_1 that is not face, contrary to assumption. Then after choose v_4 and v_5 so that v_2, v_4, v_5 are consecutive neighbours of v_1 , and collapse the triangle $v_1v_4v_5$.

This will produce no digons that are not faces, since such a digon would come before the collapsing from a triangle that is not a face inside of T_1 , contrary to assumption. There may however appear triangles that are not faces inside of T_1 . Such triangles come from quadrilaterals $v_1v_4v_6v_7$, $v_1v_5v_8v_9$, and $v_4v_5v_1v_1$. The quadrilaterals $v_1v_5v_8v_9$ are of two kinds, either containing v_4 or not containing v_4 , but may not have diagonal edges v_1v_8 or v_5v_9 , otherwise either there was a triangle that is not a face inside of the quadrilateral, or collapsing the side v_1v_5 does not give for the quadrilateral a triangle that is not a face.

This implies that all such quadrilaterals containing v_4 are pair wise contained in each other, and all such quadrilaterals not containing v_4 are pair wise contained in each other. The analogous properties hold for the quadrilaterals $v_1v_4v_6v_7$,

but these are of only one kind, namely containing v_5 , otherwise $v_6 = v_2$ and having the diagonal edge v_2v_4 . The analogous properties also hold for the quadrilaterals $v_4v_5v_10v_11$, but these are again of only one kind, namely not containing v_1 , since they are contained in the triangle $T_1 = v_1v_2v_3$. Furthermore, a quadrilateral $v_1v_4v_6v_7$ containing v_5 must contain any quadrilateral $v_1v_5v_8v_9$ not containing v_4 and must also contain any quadrilateral $v_4v_5v_10v_11$ not containing v_1 , and a quadrilateral $v_1v_5v_8v_9$ containing v_4 must also contain any quadrilateral $v_4v_5v_10v_11$ not containing v_1 . This guarantees that these quadrilaterals will not lead, after collapsing $v_1v_4v_5$, to three triangles that are not faces that do not contain each other inside T_1 , thus preserving the property that no triangle has three children.

In the remaining case for collapsing a triangle, there is a triangle T_1 that has either one child T_2 or two children T_2 and T_3 , where both T_2 and T_3 have exactly one vertex inside. Suppose T_2 shares no sides with either T_1 or T_3 . Writing $T_2 = v_1v_2v_3$, it must again consider quadrilaterals $v_1v_2v_4v_5$, $v_1v_3v_6v_7$, and $v_2v_3v_8v_9$. There may not simultaneously exist quadrilaterals $v_1v_2v_4v_5$ containing v_3 , $v_1v_3v_6v_7$ containing v_2 , $v_2v_3v_8v_9$ containing v_1 , and $v_1v_2v_4v_5$ not containing v_3 . For if $v_6 = v_5$, then $v_1v_5v_7$ is not a face and thus equals T_1 , so v_1 is a vertex of T_1 and the quadrilateral $v_2v_3v_8v_9$ cannot contain v_1 ; if $v_7 = v_4$ then $v_1v_5v_4$ is again T_1 and the same argument holds; and if $v_6 = v_4$, then $v_8 = v_5$ and $v_9 = v_7$, so the triangle $v_5v_4v_7$ is T_1 , and the quadrilateral $v_1v_2v_4v_5$ would be inside the triangle $v_1v_2v_7$ which is a face, and this is not possible.

Therefore, by symmetry, it may assume that either there is no quadrilateral $v_1v_2v_4v_5$ containing v_3 or no quadrilateral $v_1v_2v_4v_5$ not containing v_3 that will give rise to a new triangle that is not a face after identifying v_1 and v_2 . Thus if $v_1v_2v_3$ contains the single vertex v_0 , collapsing the triangle $v_1v_2v_0$ identifies v_1 and v_2 and creates only triangles with pair wise containment involving the new vertex $v_1 = v_2$, besides the triangle T_3 , thus preserving the property that no triangle has three children. If $T_2 = v_1v_2v_3$ shares one side with T_1 , say the side v_2v_3 , then one of the other two sides is not shared with T_3 , say the side v_1v_2 , and the quadrilaterals $v_1v_2v_4v_5$ cannot contain v_3 , so again it may collapse the triangle $v_1v_2v_0$ with v_0 inside T_2 , creating only triangles with pair wise containment involving the new vertex $v_1 = v_2$, besides the triangle T_3 , thus preserving the property that no triangle has three children. And if T_2 shares a side v_1v_3 with T_3 , then every quadrilateral $v_1v_2v_4v_5$ containing v_3 also contains

T_3 , so collapsing $v_1v_2v_0$ with v_0 inside T_2 gives two families of triangles with pair wise containments involving $v_1 = v_2$, one family containing v_3 and the other family not containing v_3 , again preserving the property that no triangle has three children. \square

The following proposition generalizes a result of Herbert Fleischner.

Proposition 2.2. *Suppose all red faces of G have either 4 or 6 sides, while the blue faces are arbitrary. Suppose the reduced graph H has no triangle that is not a face other than the outer triangle, and H has no digon that is not a face either. Suppose H has an odd number of vertices. Then H has a spanning tree of faces that are triangles, and so G is Hamiltonian.*

Proof. Apply Lemma 2.1 repeatedly to collapse triangle faces to single vertices while preserving the invariant property. Each step reduces the number of vertices by two, so this number remains odd until it to be left with just the outer face. The collapsed triangles form a spanning tree of faces.

This result follows from the following main observation. \square

Lemma 2.2. *Let G be as in Lemma 2.1. Suppose the reduced graph H has a triangle T that contains at least one vertex inside, such that no triangle inside of T is not a face (i.e., contains at least one vertex inside), and no digon inside of T is not a face (i.e., contains at least one vertex inside). Then finding a proper quasi spanning tree of faces for H reduces to finding a proper quasi spanning tree of faces for H' obtained from H by removing all vertices inside of T and their incident edges, and adding a parallel edge inside of T to each edge of T .*

Proof. It may proceed with a triangle T as in the preceding Lemma, and end up with either single vertex v inside of T or no vertex inside of T by repeatedly collapsing triangle faces. In the case of a single vertex v inside of T , selecting one of the three triangles involving v corresponds to selecting one of the three digons added for the sides of T in H for a quasi spanning tree of faces, and in the case of no vertex v inside of T , it may either select or not select the triangle T in H' for a quasi spanning tree of faces. The case of single vertex v inside of T is reached when T initially contains an odd number of vertices inside of T , and the case of no vertex v inside of T is reached when T initially contains an even number of vertices inside of T .

It remains to show the two cases that are used to change the parity inside of T . If there is initially a digon v_1v_2 with at least one end point inside of T , then it may select and collapse v_1v_2 , creating triangles that are not faces from quadrilaterals $v_1v_2v_4v_5$, and there are again two families of such quadrilaterals, given by the two triangle faces $v_1v_2v_3$ and $v_1v_2v'_3$, namely quadrilaterals containing v_3 and quadrilaterals containing v'_3 . Each of the two families of quadrilaterals creates triangles with pair wise containment, thus giving the property that no triangle T_1 either equal to T or inside of T has three children, satisfying the invariant property. \square

Theorem 2.1. *Let G be a 3-connected cubic planar bi partite graph. Let H be the reduced graph for G , and let H' be the sub graph of H obtained by removing all edges that do not have consecutive parallel edges. If H' has one, two, or three connected components, then H has spanning tree of faces, and thus G has a Hamiltonian cycle. The case of a single component for H' includes the case where all faces in one of the three color classes are squares.*

Proof. If H' is a single connected component, then it can choose a spanning tree of H' , corresponding to a spanning tree of digons in H .

If H' has two connected components, then it may choose a face f of H that has vertices from both components. Starting with this face f , we also consider two spanning trees of digons for the two components of H' , and add these digons one at a time as long as they do not form a cycle containing f . Eventually, the single face f and the added digons will span H . If H' has three connected components, then it may be that H has a face f touching all three components, then proceed from f as for the case for two components, by considering the three spanning trees of digons for the two components. Otherwise some component, say the first, has faces touching it and the second component and also faces touching it and the third component. Both sets of faces have at least four faces, since a cut of H has at least four edges by 3-connectivity and the fact that any cut has an even number of edges, so it may choose a face f touching the first and second component, and a face f' touching the first and third component, so that these two faces do not share any vertices. Starting with these two faces, then it again add digons from the three spanning trees for the three components so long as they do not form a cycle, until a spanning tree of faces for H is obtained.

The proof for three connected components extends to the case of four connected components, but the result does not hold in the case of five connected components.

It show next that one can decide whether the reduced graph H has a spanning tree of faces that are either digons or triangles in polynomial time. The result easily extends to the case of a spanning tree of faces where all but a constant number of faces are either digons or triangles. \square

3. GREEDY ROUTING ALGORITHM FOR SPANNING TREE OF FACES

GREEDY's sending's extraordinary bit of leeway is its dependence just on information of the sending hub's quick neighbours. The state required is unimportant and reliant on the thickness of hubs in the remote system, not the complete number of goals in the system. On systems where multi-bounce steering is valuable, the quantity of neighbours inside a hub's radio range must be generously not exactly the all out number of hubs in the system. The position a hub partners with a neighbour turns out to be less current between reference points as that neighbour moves. The precision of the arrangement of neighbours additionally diminishes; old neighbours may leave and new neighbours may enter radio range.

Consequently, the right decision of beaconing interim to keep hubs' neighbour tables current relies upon the pace of portability in the system and scope of hubs' radios. It demonstrates the impact of this interim on GPSR's presentation in our reproduction results. By keeping current topological state for a one-bounce range about a switch is the base required to do any directing, no helpful sending choice can be made without learning of the topology at least one jumps away.

Negligible Depth Spanning Tree calculation frame tree is a spreading over tree where every hub has a related curved structure that contains the areas of all its relative hubs. Structure trees give a method for conglomerating area data and they are worked by amassing curved frame data up the tree. Data is utilized in steering to keep away from ways that are not gainful; rather than cross a fundamentally diminished sub tree, comprising of just the hubs with arched bodies containing the goal point. Every hub in fundamental structure tree stores

data about the curved frames that contain the directions of the considerable number of hubs in sub trees related with every one of its youngster hubs.

The information of the convex hull is based on the aggregation of the tree. Here the convex hull will be computed for coordinating and communicating the nodes. In entire network the root node is associated with all the nodes in convex hull. This convex hull consists of polygon where minimum points are obtained. the explanation of minimum depth spanning tree is given below:

- (1) Minimal depth spanning tree: this will determine the neighbour nodes using the smallest number of hop in the root. Here one root is present in entire system, then that node is known as parent node. Similarly, more than one node is present in the tree then the node is selected based on distance of node n. the minimal depth spanning tree is closely related to the path of minimal spanning tree. The below shows the explanation of minimal path spanning tree.
- (2) Set of neighbour nodes are obtained by determining the path of minimal spanning tree. The length of the root also obtained while determining the path. Below are some relations for determine the path of spanning tree:
 - Here geometric distance is calculated if there is more than one node in the entire set.
 - Extreme nodes will produce minimum length of the spanning tree.

Here a packet is used to transverse the expected value of tree. Here the both length and number of hop is calculated. Based on the diameter of network the routing performance is calculated. The proposed routing algorithm gives low density but in lemma and lemma 2, the density will be high. Hence this algorithm gives effective output.

Basically, the local spanning trees will give effective output compared to global spanning tree.

4. IMPLEMENTATION OF ALGORITHM

The below Figure 1 shows the block diagram of proposed algorithm. Here first greedy mode is implemented. Next spanning tree provides operation in effective way. Here tree mode is implemented to provide proper routing. Proper routing is maintained by introducing the greedy routing. The spanning tree will

perform the operation in effective way. At last target is set for the operation that is performed.

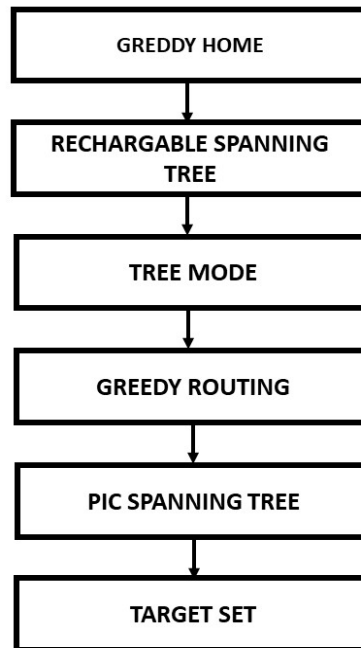


FIGURE 1. Flow chart of greedy routing algorithm

- (1) Step 1: Check for Greedy Mode: If $p: mode = \text{Polynomial Spanning}$, follow step 6.
- (2) Step 2: Check Reached spanning Tree: If the root has a node with a convex hull which intersects with R , follow step 5. Otherwise, follow step 3.
- (3) Step 3: Find Tree Mode: If $p: mode = \text{Find Tree}$: If v is the root node for $p: \text{Tree}$, algorithm terminates here. Otherwise, forward p to the parent node in $p: \text{Tree}$.
- (4) Step 4: GREEDY Routing: Route packet to destination t according to Algorithm. If packet is undeliverable, set $p: mode = \text{Find Tree}$ and follow step 3.
- (5) Step 5: Pick spanning Tree for providing routing for faces: If R is contained in either of the grid squares of the local hull trees, set $p: \text{Tree}$ as

the local tree (in a grid square that contains R) with a root that is closest to the t . If the grid squares of the local hull trees do not completely contain R , set p : Tree as the global tree with a convex hull that contains R ; if such a global tree does not exist, pick the global tree with a root that is closest to t . Follow step 6.

- (6) Step 6: Greedy routing to Target Set: Determine target set B for message broadcast with respect to p : Tree according to the following rules: If p : mode = greedy, the node from which greedy message was originally received is not to be included in set of targets. If p : Tree is a local tree, each neighbouring node that has an associated convex hull (from v 's perspective) that intersects R is added to the target set.

5. CONCLUSION

In this paper, the greedy routing algorithm for Hamiltonian cycle from quasi spanning tree of faces is implemented. Here issues routing is applied to the spanning tree. The proposed algorithm is used in various practical applications. The proposed algorithm is suited to both minimum face spanning sub graph and maximum space spanning sub graph. The entire tree structure is based on the sub graph. A lower bound for both the problems based on the number of vertices which is tight. A tight upper bound of the number of vertices of a minimal face-spanning sub graph. However, to design Routing algorithms with better approximation ratio for the face-spanning sub graph problem and the minimum-vertex face-spanning sub graph problem are left as open problems. Hence the proposed algorithm provide routing to detect the problems and analyse it.

REFERENCES

- [1] G. BRINKMANN, J. GOEDGEBEUR: *Generation of cubic graphs and snarks with large girth*, Journal of Graph Theory, **86**(2) (2017), 255–272.
- [2] G. BRINKMANN, K. COOLSAET, J. GOEDGEBEUR, H. M'ELOT: *House of Graphs: A database of interesting graphs*, Discrete Applied Mathematics, **161**(2013), 311–314.
- [3] G. BRINKMANN, J. GOEDGEBEUR, B. D. MCKAY: *Generation of cubic graphs*, Discrete Mathematics and Theoretical Computer Science, **13**(2) (2011), 69–80.
- [4] S. ABBASI, A. JAMSHED: *A Degree Constraint for Uniquely Hamiltonian Graphs*, Graphs and Combinatorics, **22**(2006), 433–442.

- [5] J. M. BOYER, W. J. MYRVOLD: *On the Cutting Edge: Simplified $O(n)$ Planarity by Edge Addition*, Journal of Graph Algorithms and Applications, **8**(3) (2004), 241–273.

DEPARTMENT OF MATHEMATICS
JNTU COLLEGE OF ENGINEERING ANANTAPUR
E-mail address: vijaypachalla@gmail.com

DEPARTMENT OF MATHEMATICS
JNTU COLLEGE OF ENGINEERING ANANTAPUR