Finding fundamental cutsets in a Graph

Akshat Aggarwal
IIM2014005
7th Semester Dual Degree (IT)

Yogesh Gupta
IRM2014004
7th Semester Dual Degree (IT)

Vishesh Middha
ISM2014007
7th Semester Dual Degree (IT)

***Abstract*—In general, cut-sets can be used to identify weak spots in a graph. So finding cut-sets can be useful in many cases. However, a graph can have large number of cut-sets. So, similar to the strategy of using fundamental circuits to find all other circuits, we can use combination of two or more fundamental cut-sets to find all cut-sets of a graph. As a result, in this assignment we strive to find the fundamental cut-sets of a graph.**

***Index Terms*—graph, cutsets, fundamental cutsets, spanning trees**

I. Introduction

We consider an undirected graph $G = (V, E)$ where,

- V is a set of vertices (or nodes), and
- $E \subseteq (V \times V)$ is a set of edges.

Our objective is that given a graph G, find out the fundamental cut-sets of the graph. After introducing the necessary notation and basic definitions, we explain our approach of finding the fundamental cut-sets.

A. Definitions and Notations

1) **Cut** - In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets.

2) **Cut-set** - A cut set is a minimal set of edges that, when broken, breaks the graph into two completely separate parts (two groups of nodes). Any cut determines a cut-set, the set of edges that have one endpoint in each subset of the partition. These edges are said to cross the cut. In a connected graph, each cut-set determines a unique cut, and in some cases cuts are identified with their cut-sets rather than with their vertex partitions.

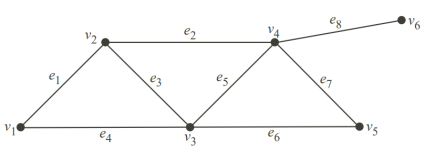
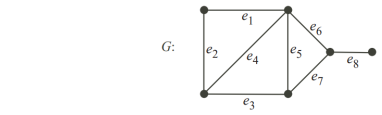
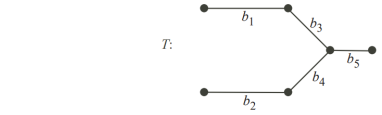


Fig. 1: $\{e_1, e_4\}, \{e_6, e_7\}, \{e_1, e_2, e_3\}, \{e_8\}, \{e_3, e_4, e_5, e_6\}, \{e_2, e_5, e_7\}, \{e_2, e_5, e_6\}$ and $\{e_2, e_3, e_4\}$ are the cut-sets of above graph

3) **Fundamental cut-set** - A fundamental cut-set with respect to that tree is a cut set that only contains one branch of the tree. There may be many fundamental cut sets with respect to a given tree.



has the spanning tree



that defines these fundamental cut sets:

$b_1 : \{e_1, e_2\}$ $b_2 : \{e_2, e_3, e_4\}$ $b_3 : \{e_2, e_4, e_5, e_6\}$
$b_4 : \{e_2, e_4, e_5, e_7\}$ $b_5 : \{e_8\}$

Fig. 2: Listing fundamental cut-sets

II. Motivation

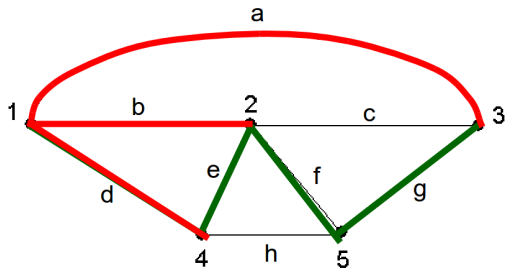
Graphs are very important data structure, finding uses in numerous applications such as in network related algorithms, routing, finding relations among objects, shortest paths and many more real life related applications. On e-commerce websites, relationship graphs are used to show recommendations. Connecting with friends on social media, where each user is a vertex and the connection between two users is represented by an edge, is also an important application of graphs.
Cut-sets are of great importance in studying properties of transportation and communication networks. Suppose , for example that there are six cities connected by telephone lines [1] and we need to find out if there are any weak spots. We can do so by finding the smallest cut-set. To find all cut-sets can take exponential time, So, we can first find the fundamental cut-sets(Definition 3) and use combination of them to find all other cut-sets.

III. Methods and Algorithms

A fundamental cut-set has meaning only with respect to a given spanning tree. Each branch of the spanning tree defines a fundamental cut-set.
So, for our purpose we consider any one spanning tree of the graph which be found in linear time using an algorithm such as Breadth-first or Depth-first search.
We then find out the fundamental cut-sets of the

graph with respect to a spanning tree. For this, we follow a straight-forward approach consisting of the following steps -

1) Get a spanning tree. Separate the edges of the graph into branches(those that are part of the spanning tree) and chords(those edges that are not part of the spanning tree).
2) For each branch of the spanning tree, remove it. Then we get 2 disconnected components.
3) Find all edges connecting these two components.
4) These edges along with the branch form a cut-set.
5) This way we find a fundamental cut-set for each branch of the spanning tree.



A simple procedure for above steps is given in Algorithm 1.

Algorithm 1 Finding fundamental cut-sets in a graph with respect to a spanning tree of the graph

Input: Graph G, Spanning Tree T
Output: List of fundamental cutsets in G with respect to a spanning tree, T

$v \leftarrow$ number of vertices in a graph
$e \leftarrow$ number of edges in a graph
$b \leftarrow$ number of branches in spanning tree T
$r \leftarrow m - b$, number of chords
for each branch b_i **do**
  Let u, v be the two vertices in spanning tree T, connected by branch b_i
  Remove branch b_i from T
  Let V_1, V_2 be the two set of vertices formed after removing b_i
  $cutset \leftarrow \{\}$
  for each chord c_i **do**
    if c_i connects two vertices v_1, v_2 such that one belongs to V_1 and the other to V_2 **then**
      Add c_i to $cutset$
    end if
  end for
  $cutset \leftarrow cutset \cup b_i$
  Add $cutset$ to $result$
  Add branch b_i to the T
end for
return $result$

IV. Results and Conclusion

First, we found a spanning tree of the given graph. Time complexity of doing this $O(V + E)$.

Then, for each branch, we first divide the vertex set into two sets V_1, V_2. This can be done in $O(V)$ time. Then we check for each chord if it connects two vertices from different sets. If yes then we add that chord to cut-set. Going through all chords takes $O(E - V + 1)$ time. So for each branch we do processing which takes $O(V + E - V + 1)$ or $O(E + 1)$ i.e $O(E)$.

There are $(V - 1)$ one branches in the spanning tree. So overall time complexity is $O(V \times E)$.

Below is the plot of time complexity vs size of input :-

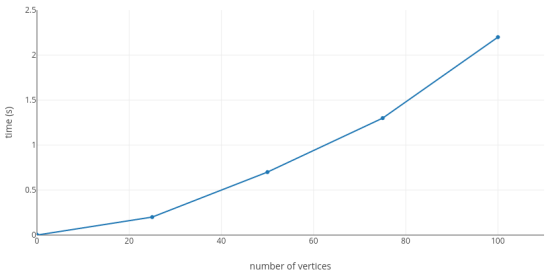


Fig. 3: Time vs input size

References

[1] Graph Theory with Applications to Engineering and Computer Science, Narsingh Deo.