An Optimization Algorithm for Finding Graph Circuits and Shortest Cyclic Paths in Directed/Undirected Graphs

Rajesh R. Raina, Dr. Rakesh K. Katare, Shazad A. Mughal, Manjula Dwivedi

Abstract— Computing combined circuits and shortest cyclic paths between two given nodes in undirected graphs is a fundamental operation over graphs. While a number of techniques exist for answering computing circuits and approximating node distances efficiently in directed graphs, but the actual circuit calculations, their generations and shortest cyclic paths in undirected graphs are often neglected. However, it is often essential to find out all combined circuits and shortest cyclic paths between two given nodes in an undirected graphs. In this paper, we have addressed this problem and presented an optimistic algorithm that not only supports calculation of circuits but also generates combined circuits and computes corresponding shortest cyclic paths in undirected graphs. This algorithm is also applicable to directed graphs as well.

Index Terms— Algorithm, Graph Theory, Shortest Cyclic Paths, Combined Circuits, Explored Graphs (Tree based).

1 INTRODUCTION

A graph G is defined as a set of two tuples that is G= (V, E), where V represents set of vertices of G and E represents the set of edges of G. There exists a mapping from the set of edges to a set of pairs of elements of V. An edge is denoted by the (unordered) pair of its endpoints. Each edge e of a graph is weighted or un-weighted with a real number w (e), which extends to a weight function on all sets of edges. Weights are allowed to be negative, but the weight of a simple circuit (a connected sub-graph which is regular of degree 2) cannot be allowed to be negative, as in the shortest path problem.

A path in a graph is a sequence of edges which connect a sequence of vertices and a cycle/circuit is a path such that the start vertex and end vertex are the same. Cyclic paths or circuits in a graph is a graph that contains n number of edges and vertices in a closed chain where the no. of edges in circuits is equal to the no. of vertices with degree 2, that is every vertex has exactly two edges incident with it. A path is elementary if no vertex appears twice. A circuit is elementary if no vertex but the first and last appears twice. Two elementary circuits are distinct if one is not a cyclic permutation of the other. There are c distinct elementary circuits in G. Hamiltonian cycle, Hamiltonian circuits is a cycle that visits each vertex exactly once (except for the vertex that is both the start and end, which is visited twice). A graph that contains a Hamiltonian cycle is called a Hamiltonian graph. A Hamiltonian decomposition is
an edge decomposition of a graph into Hamiltonian circuits. For example, a complete graph with more than two vertices is Hamiltonian.

Therefore every cycle graph is Hamiltonian and every circuit or Hamiltonian cycle can be converted into path by removing one of its edges but a Hamiltonian path can be extended to Hamiltonian circuit only if its endpoints are adjacent [5], [6].

2 OBJECTIVES

Broadly speaking, there are two enumeration problems on sets of objects. The one, which we call counting, is determining how many objects there are in the set. The other, which we call finding, is the construction of every object in the set exactly once. Indeed, objects may always be counted by finding them if a method to do so is at hand. But knowing the count is usually of little aid in finding the objects.

The objective is to find the total number of combined elementary circuits, their formation and shortest cyclic paths of an undirected/directed graph which could be faster than the algorithms previously known. Specific counting problems are, of course, solved. For example, there are exactly

$$\sum_{i=1}^{n-1} \binom{n}{n-i+1} (n-i)!$$

elementary circuits in a complete directed graph with n vertices. Thus the number of elementary circuits in a directed graph can grow faster with n than the exponential 2n, so in the complete undirected graph. Therefore it is clear that our aim is to reduce the total no. of Circuits formed in a graph whether directed or undirected with n vertices, e edges and c elementary circuits by some integer number ‘k’ which will be

- R. R. Raina is with the Department of Computer Science, A.P.S. University, Rewa (M.P.)-India-486003. E-mail: luckyrajmn@gmail.com, rajesh_roshan14mn@yahoo.co.in.
- Dr. R. K. Katare is with the Department of Computer Science, A.P.S. University, Rewa (M.P.)-India-486003. E-mail: katare_rakesh@yahoo.com, katare1962@gmail.com.
- S. A. Mughal is with the Department of Computer Science, A.P.S. University, Rewa(M.P.)-India-486003. E-mail: shazadmughal47@gmail.com.
- M. Dwivedi is with the Department of Computer Science, A.P.S. University, Rewa (M.P.)-India-486003. E-mail: manjulaaa123@gmail.com

ISSN 2229-5518

feasible for a substantially larger class of problems than the best algorithms previously known [1], [2], [3], [4], realizes a time bound of $O((n + e)(c + 1))$,$O(n.\ e(c + 1))$,$O(e^3\ v)$, etc. and are applicable only to directed graph.

Once the combined circuits have been developed, the next aim is to compute the cyclic paths in such a way which will lead to the shortest cyclic paths in the graph.

3 Literature Review

The literature contains several algorithms which find the elementary circuits of any direct graph. In the algorithms of Tiernan [7] and of Weinblatt [8], time exponential in the size of the graph may elapse between the output of one circuit and the next [2]. Tarjan [2] presents a variation of Tiernan's algorithm in which at most $O(n\ .\ e)$ time elapses between the output of any two circuits in sequence, giving a bound of $0(n\ .\ e(c + 1))$ for the running time of the algorithm on an entire graph in the worst case. Ehrenfeucht, Fosdick, and Osterweil [3] give a similar algorithm which realizes the same bound. In the case of Tarjan's algorithm, the worst-case time bound is realized. We assume that the algorithm begins with vertex 1 and, in any search from vertices 1 through k + 1, it visits vertices k + 2 through 2k + 1 before a first visit to vertex 2k + 2. In the course of finding each of the k elementary circuits which contain vertex 1, the subgraph on vertices 2k + 2 through 3k + 3 will be explored k times, once for each of the vertices k + 2 through 2k + 1. Thus exploration from vertex 1 alone consumes 0(k3) time. Since there are exactly 3k elementary circuits in the entire graph, the running time is at least $O(n\ .\ e(c + 1))$. Johnson[1] presented an algorithm, which finds all the elementary circuits of a directed graph in time bounded by $O((n + e)(c + 1))$ and space bounded by $O(n + e)$, where there are n vertices, e edges and c elementary circuits in the graph. The algorithm resembles algorithms by Tiernan and Tarjan, but is faster because it considers each edge at most twice between any one circuit and the next in the output sequence.

To find the basis of the cycle space of a graph with minimum total weights, Steeves [9] and Cribb, Ringeisen and Shier[10] may have some uses in Surveying and algorithms have been developed in [9] and [11]. Hubicka and Syslo [11] conjectured that their algorithm works, but Kolasinska has recently constructed a counterexample [12]. Steeves in [9] developed an algorithm that takes $0(e\ v^2)$ operations, but counterexamples have also been found for it as well.A second use for minimum cycle bases may be to improve algorithms that list all simple circuits in a graph. One early reference and one recent reference for this type of algorithm are [13] and [14]. Dixon and Goodman use a similar technique to search for the longest cycle in a graph [15]. Horton [4] gave an algorithm that solves the above problem in $O\ (e^3\ v)$ operations.

4 Proposed Research Work

With the basic background covered under the above, the need of such algorithm is required that can be used as a quick trick to produce combined elementary circuits and shortest cyclic paths in a graph. Although there are many algorithms available but they require sophisticated computing. Therefore the proposed research work attempts to reduce such sophisticated computing and this will be done by developing an optimistic algorithm to calculate and generate combined elementary circuits and shortest cyclic paths in any graph. The proposed algorithm reduces 'k' number of combined circuits from the total number of circuits formed in a graph which also leads to save nearly 't/2' time (i.e., if 't' is total time to calculate circuits). Circuit calculation method will help us in finding the shortest cyclic paths by considering weighted graphs. The proposed research work done this by finding the appropriate circuit, compute their weights and compare with each other, the lowest will be the shortest path or shortest cyclic path of the graph. Through Circuits, the proposed research work can show the robustness, performance, etc. of the network architecture. We can also show the maximum and minimum cycle length. In cyclic paths or circuits, the choice of the start vertex is arbitrary, therefore every node/vertex can be shown as a full duplex node or in other words, works under the client server architecture which means any vertex can receive any data from any other vertex or sends to any other vertex contains within the cycle. The proposed work also helps us in finding other different cycles by interchanging the vertices, for example to find the circuit for 'x' vertex, then swap each 'x' vertex with the each 'y' vertex (where let 'y' being the root/start vertex or the source/ destination vertex) in the proposed work. The proposed work also finds the total number of circuits formed in a graph.

5 Algorithm Description

In this section we explain our algorithm for finding graph circuits and shortest cyclic paths for undirected graphs in detail. The algorithm presented in this paper in turn can be applied to directed graphs as well. Let G={V,E} be a complete undirected graph as shown in Fig. 1, where V is the set of vertices {0,1,2,3} and E is the set of edges. In this case V=4 and E=10. P= $x_1e_1x_2e_2 \ldots x_n$ for $x_i \in V\ (G)$, $e_i \in E\ (G)$, $1 \leq i \leq n$, is called a path of G if all vertices x_i in P are distinct, and x_1 is called origin vertex, x_n is called the terminus vertex. A cycle is defined as a path except that x_1=x_n. Circuits with only one edge is not considered to be a circuit in this paper. Fig. 2 is the exploration of complete undirected graph (Fig. 1) includes node number and level number. Exploration of graphs can be done by considering any node as the source or initial node and follow the following rules as mentioned in the given below alg

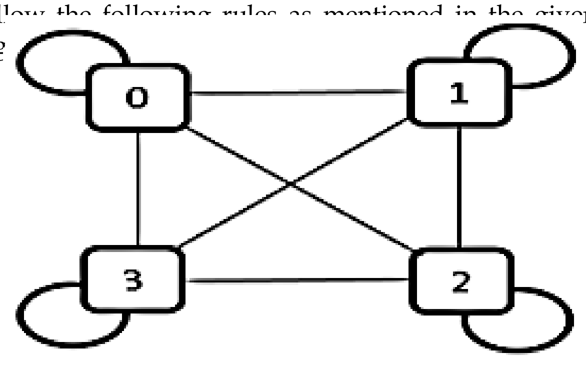


Fig. 1. Complete undirected graph

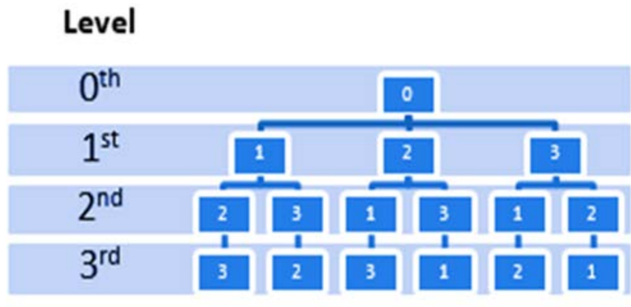


Fig. 2. Exploration of Fig. 1 into a tree.

Once the exploration is done, next is to generate the circuits which can be calculated as well. This way we can generate and calculate the combined circuits which in turn can be used to generate and calculate the other circuits by simply exchanging the nodes. Once all circuits are formed, we can find out the shortest cyclic paths of weighted graphs. Therefore the shortest cyclic path of length 'l' with weighs 'w_i' from u to u, denoted by dist(u, u), is the minimum summation of edges/weights or zero if not u Є V (G):

$$\text{dist(u,u)} = \begin{cases} \min\left(\sum_{i=1}^{n}(W_i)\right) & \text{if } u \in V(G) \\ 0 & \text{else} \end{cases} \quad (1)$$

Following is the algorithm which helps us in exploring any graph (directed or undirected) into tree based, generation of combined circuits, calculation of circuits and shortest cyclic paths.

Algorithm:

Step 1. Begin

Step 2. Explore graph into tree based from left to right in such a way that the descents should not be repeated in a particular path. Exploration is done on the basis of existed communications in the graph.

Step 3. The parent node i is at level i-1 while as the leaf node is at level n-1, where 1 ≤ i ≤ n-1.

Step 4. To form circuits of length j where j ≤ n , traverse the path of descents from source node i at level i-1 to the nodes at level j-1 from right to left fashion and goto Step 5 & 6 until reach to the last left sibling node. This comparison continues until no ancestor sibling node on the left side.

Step 5. Compare every child node (terminal node) at level j-1 with the left sided sibling nodes of the ancestor node at level i.

Step 6. If a match found, store/display the path/circuit else choose next path and repeat step 3.

Step 7. End.

Step 8. Exit.

TABLE 1
COMBINED CIRCUITS FOR VERTEX '0'

Length = 4	Length = 3	Length = 2
0-3-2-1-0	0-3-2-0	0-3-0
0-3-1-2-0	0-3-1-0	0-2-0
0-2-3-1-0	0-2-1-0	0-1-0

Using the above algorithm, Fig. 2 can be constructed from Fig. 1. Therefore according to the above algorithm, resulted combined circuits can be constructed as shown in table 1.

From the above table it is clear that the minimum length is 2 and the maximum length of the combined circuit for vertex '0' is 4. The total number of combined circuits for vertex '0' is 9, therefore the total number of combined circuits for four vertices is equal to 36.

The comparisons between the vertices of the explored graph can be better understand with the help of Fig. 3 and Fig. 4 is the explored graph of Fig. 3.

Also we have derived some useful lemmas by studying the topological properties of the explored graph are as under:

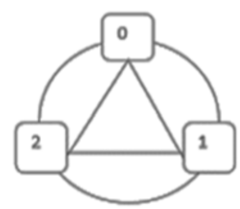


Fig. 3. Complete undirected graph with three vertices.

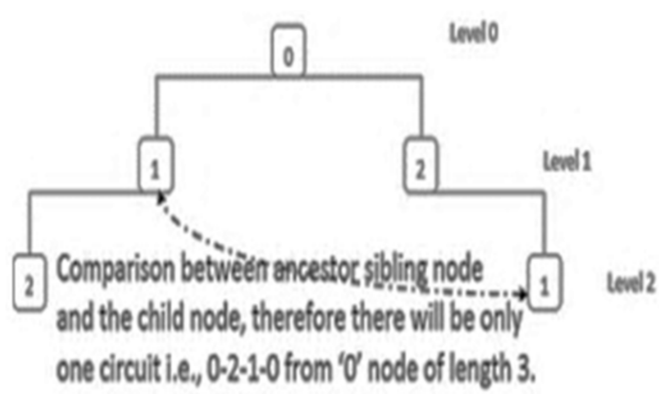


Fig. 4. Explored graph of Fig. 3.

ISSN 2229-5518

Lemma 1. *Explored graph can have (n-1) levels only.*

Proof. Let n_i be the total number of nodes/vertices in the graph. The source node or the parent node of the explored graph (tree) is at 0th level. Therefore the last node or the leaf node in a combined circuit can be at $(n-1)^{th}$ level which is the last level of the explored graph.

Lemma 2. *Total number of combined circuits at level i where 0<i<n, is equal to some 'y' number of matches found in explored graph.*

Proof. Let 'x' be the total number of matches found in the explored graph at level 'l_i' , $0 < i < n$. Then their exists some 'y' comparisons that constitutes combined circuits. Hence the total number of combined circuits at level 'l_i' that leads to constitute combined circuits is 'y'.

Lemma 3. *Total number of combined circuits in explored graph is equal to the*

$$n\sum_{i=1}^{n-1}(\text{Total no. of circuits at level i})$$

Proof. Using Lemma 2, it is clear that there are 'y' number of combined circuits at level 'l_i', where $0 < i < n$. since the total number of level are (n-1).

$$n\sum_{i=1}^{n-1}(\text{Total no. of circuits at level i})$$

Therefore, total number of combined circuits for n vertices is

Lemma 4. *The shortest cyclic path (dist(u,u)) from u to u of length 'l', weight 'w_i' is:*

$$\text{dist(u,u)} = \begin{cases} \min\left(\sum_{i=1}^{n}(W_i)\right) & \text{if } u \in V(G) \\ 0 & \text{else} \end{cases}$$

Proof. Let x_i where $0 < i \le n$ be the weight / distance of weighted graph and let w_i where $0 < i \le n$ be the weights of their combined circuits. Therefore the shortest cyclic path dist(u,u) from u to u is the minimum weight of length l_i where $0 < i \le n$, i.e,

$$\text{dist(u,u)} = \begin{cases} \min\left(\sum_{i=1}^{n}(W_i)\right) & \text{if } u \in V(G) \\ 0 & \text{else} \end{cases}$$

Lemma 5. *'t/2' time is saved in the above proposed algorithm as compared to the previously known algorithms.*

Proof. Let 't' is the total time consumed for calculating circuits using combination permutation technique and which can be calculated as ${}^{n}C_{r} = \frac{n!}{(n-r)!r!}$ where 'n' is the number of nodes or

vertices and 'r' is the length of the circuit. Let n=4 and r=4, then,

$${}^{n}C_{r} = \frac{4!}{(4-4)!4!} = \text{Infinity}$$

i.e., infinity time is used while as in the above algorithm it is only 12 circuits as shown in Fig.2.

Lemma 6. *The maximum length of the combined circuit is 'n'.*

Proof. Since there is no repetition of vertices or nodes from u to u, therefore the maximum length of the combined circuit is 'n' as there are 'n' number of vertices in the graph.

Lemma 7. *The minimum length of the combined circuit is 2 for n >1.*

Proof. Since each U_i is connected to each V_i in an undirected complete graph, $0 < i \le n$, then there is a direct path from U_i to V_i and from V_i to U_i. As circuits of length one is not considered in this paper, therefore the minimum length of the combined circuit is 2 for $n > 1$

6 Conclusion

In this paper we have presented an optimistic algorithm for generating and calculating combined circuits of the undirected graphs. The algorithm in this paper can be applied to any kind of graph or topological network. The fundamental task over graphs is to find out the accurate shortest cyclic path from U_i to U_i of length l_i, $0 < i \le n$ and $n > 1$, is also presented in this paper. Formation of combined circuits shows the robustness and performance of the network. The more combined circuits, the more robust is the network and hence performance. The proposed algorithm in this paper doesn't produce repetitive combined circuits when used for directed or undirected graphs and hence saves much of the time. As such we have studied topological properties of the explored graph and developed some very useful lemmas. We hope that these lemmas will be our guidelines for the further study in the field of computer science. We have also shown the maximum and the minimum length of the combined circuits through these lemmas.

References and Bibliography

[1] D.B. Johnson, SIAM J. Comput. 4, 77 (1975)

[2] R. TARJAN, Enumeration of the elementary circuits of a directed graph, this Journal, 2 (1973), pp. 211-216.

[3] A. EHRENFEUCHT, L. FOSDICK AND L. OSTERWEIL, An algorithm for finding the elementary circuits of a directed graph, Tech. Rep. CU-CS-024-23, Dept. of Computer Sci., Univ. of Colorado, Boulder, 1973.

[4] J.D. Hortan, A Polynomial-Time Algorithm to Find the Shortest Cycle Basis of a Graph, TR84-026, School of Computer Science University of New Brunswick FREDERICTON, N.B. E3B 5A3, 1984.

[5] www.wikipedia.com

[6] www.google.com

[7] J. C. TIERNAN, An efficient search algorithm to ,find the elementary circuits of a graph, Comm. ACM, 13 (1970), pp. 722-726.

[8] H. WEINBLATT, A new search algorithm for finding the simple cycles of a finite directed graph, J. Assoc. Comput. Mach., 19 (1972), pp. 43-56.

[9] P.A. Steeves, "Numerical Processing of Horizontal Control Data-Economization by Automation", unpublished Ph.D. thesis, University of New Brunswick, 1984.

[10] D.W. Cribb, R.D. Ringeisen, and D.R. Shier, "On Cycle Bases of a Graph," Congressus Numerantium, 32{1981), 221-229.

[11] E. Hubicka and M.M. Syslo, "Minimal Bases of Cycles of a Graph", Recent Advances in Graph Theory, Proceedings of the Symposium held in Praque, June 1974, Academia Praha, Praque, 1975.

[12] E. Kolasinska, "On a Minimum Cycle Basis of a Graph", Zastosowania Matematyki, 16{1980), 631-639.

[13] J.T. Welch Jr., "A Mechanical Analysis of the Cyclic Structure of Undirected Graphs", J. ACM, 13{1966), 205,210.

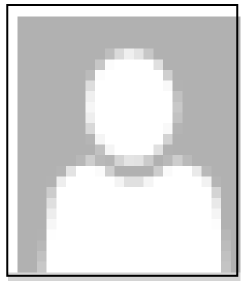
[14] M.M. Syslo, "An Efficient Cycle Vector Space Algorithm for Listing All Cycles of a Planar Graph", SIAM J. Comput., 10(1981), 797-808.

[15] E.T. Dixon and S.E. Goodman, "An Algorithm for the Longest Cycle Problem", Networks, 6(1976), 139-149.

Rajesh R. Raina did his B.Sc. (Non-medical) in 2005 from Amar Singh College Srinagar, Master of Computer Application (M.C.A.) from IUST Awantipora in 2010, and Master of Philosophy in Computer Science from APS University Rewa (MP) in 2012. He has an experience of 2 year as College lecturer. He is currently working as a lecturer in GDC Tral and is a member of different communities. His area of interest is Ph.D. in Computer Science with number of published papers in reputed journals and conferences.

Dr. Rakesh Kumar Katare did his Ph.D. in Computer Science from School of Computer Science, Devi Ahilya University, Indore (M.P.) India in 2007. He is Porfessor in Department of Computer Science, A.P.S. University, Rewa (M.P.) 486003 India. M.Tech. in Computer Science from the Institute of Computer Science and Electronics, Devi Ahilya University, Indore (M.P.) India, in First Division with Distinction. M.Phil. and M.Sc. in Physics from Dept. of P.G. studies and Research in Physics, R.D. University, Jabalpur (M.P.) India, in 1986 and 1985 respectively, in First Division. He has list of publications and Journals.

Shazad Ahmed Mughal did B.Sc. (Medical) from Jammu University in 2006, M.C.A. from BGSB University, Rajouri in 2009, M.Phil. from APS Univeristy, Rewa, M.P.-India in 2012. He is currently pursuing Ph.D. in computer science from BGSB University, Rajouri. His area of interest is Data Mining and Data Warehouse.



Manjula Dwivedi did her B.Sc. from APS University, Rewa, M.P. India in 2006, M.Sc. IT from Shri RamaKrishna, Satna, M.P. in 2008 and Currently pursuing M.Phil. from APS University, Rewa, M.P. India. She has three years of experience as Assistant Professor in Computer Science at Sindhi Girls College, Satna, M.P. India. Her area of interest is programming and algorithms.