IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-9 E-ISSN: 2347-2693

An Algorithm with Optimal Time and Space Efficiency to Detect Balance of a Signed Graph

Sashi Kanth Reddy Avula[1*], P. Siva Kota Reddy[2]

[1] Research Scholar, Department of Computer Science, JAIN University, Bangalore

[2] Department of Mathematics, Siddaganga Institute of Technology, Tumkur

Available online at: www.ijcseonline.org

Received: 19/Aug/2016 Revised: 29/Aug/2016 Accepted: 14/Sept/2016 Published: 30/Sep/2016

Abstract: A Signed Graph H = (G, σ) where G is a graph G = (V, E) and σ is a sign function σ =(+, -). When talk about a signed graph, the main focus is on its balance. In this paper we propose a dynamic programming algorithm based on depth-first search graph traversing which detects the balance in the signed graph. The proposed algorithm traverses every edge of the input signed graph at most once. The implementation of the proposed algorithm will use two linear lists (arrays), one of length |V| and the other of length (|V|+1)/2. Hence the algorithm will work much efficiently with respect to time complexity and space complexity.

Keywords: Signed Graph, linear lists, dynamic programming, time complexity, space complexity, balance.

1. Introduction

A graph represented with G = (V, E) where V is the vertex set V = {v_1, v_2, v_3,…. } and E is the edge, the line joining two distinct vertices V, E = {e_1,e_2,e_3,...}. A graph G with each of its vertex and edge assigned with a sign σ = {+, -} is said to be signed graph (also known as Sigraph). The sign of an edge is same as the product of its vertices.

$$\sigma (E_1) = \sigma (V_1) * \sigma (V_2)$$

A Sigraph is represented as H = (G, σ) where G = (V, E) is the graph and σ = (+, -) is a sign function. A Sigraph is said to be balanced if every cycle in it have the positive sign. To determine the sign of the cycle in a given Sigraph, the product of all the edges in the specific path is considered.

$$\sigma (Cycle) = \sigma (E_1) + \sigma (E_2) + \sigma (E_3) + \ldots\ldots + \sigma (E_i)$$

All the notions of the signed graph are quoted in Harary. F [1],[2],[3]. The balance factor of the signed graph was first used by Heider in his work with three vertices [4]. There are numerous applications on the balance of the signed graph. In his paper, Harary F had given the characterization of a signed graph as follows which is also referred to as 'Structure Theorem' [2]:

Theorem: Let H = (V, E, σ) be a signed graph. Then the following conditions are equivalent:

i. *H is balanced.*
ii. *Every cycle in H is positive, i.e., has an even number of negative lines.*
iii. *Any two paths between every pair of disjoint vertices of V(G) have the same sign.*
iv. *The set V(G) can be partitioned into two disjoint sets called coalitions, one of which may be empty, such that every positive line joins vertices of the same coalition and every negative line joins vertices of different coalitions.*

E. Loukakis had proposed a dynamic programming algorithm to test the balance of a signed graph based on condition (iii) of the Harary's Structure Theorem [1]. In his paper, Loukakis proved that the algorithm using three linear arrays and Breadth-First search technique has the improved time complexity and space complexity [1].

This paper is systematized as follows: We have given a brief introduction to the Signed graph and its behavior. Here in the introduction, we also gave the theorems proposed earlier on the signed graphs and its balance factors. In this paper, we propose an algorithm based on condition (iii) only and uses Depth-First search technique for graph traversing. Here each edge of the input graph is traversed at most once. The proposed algorithm uses only two linear lists (arrays) one of length |V| and the second list of length (|V|+1)/2 only. Since the algorithm proposed in this paper requires only two linear arrays and also on array with much less size, the algorithm will give much better results with respect to time and space complexities when implemented on computer.

2. The algorithm for Balance Detecting

Based on Heider [4], we consider that the input graph is a connected signed graph. The proposed algorithm is based on condition (iii) of Structure Theorem.

According to condition (iii), any two paths with every pair of different vertices of G should have equal number of negative edges. As highlighted in the proposed algorithm, here we adopt the Depth-First search graph traversing technique. Depth-First search graph traversing technique has the upper hand with respect to space complexity than the Breadth-First Search technique. The time complexity of the algorithm with both Depth-First search technique and Breadth-First search technique will be same provided the graph is represented by the adjacency list structure [5,6,7,8,9].

A connected signed graph can be better represented using an adjacency matrix. When we implement the algorithm on computer, it requires a two dimensional array to read the connected signed graph. The two dimensional array occupies more space compared to the linear array. Also the two dimensional array requires more time to process data than that of a linear array.

Algorithm 1: Signed_Graph_Balance_Detect (H, N, σ)

```
1.  BEGIN
2.      DISCOVERED(N) = true
3.      Balance = true
4.      σ [N] = + or -
5.      For each p ∈ V(G) - {N}
6.          DISCOVERED[p] = false
7.      END For
8.      top = -1
9.      For each p ∈ Adj[N]
10.         σ[p] = σ[N] * σ[(N, p)]
11.         top = top + 1
12.         Stack[top] = p
13.         DISCOVERED[p] = true
14.     END For
15.     While ( (top != -1) && (Balance == true) )
16.         p = Stack[top]
17.         top = top - 1
18.         DISCOVERED[p] = true
19.         For each q ∈ Adj[p]
20.             IF DISCOVERED[q] = false Then
21.                 DISCOVERED[q] = true
22.                 top = top +1
23.                 Stack[top] = q
24.                 σ[q] = σ[p] * σ[(p, q)]
25.         ELSE
26.                 IF (σ[p] * σ[(p, q)] != σ[q] ) Then
27.                     Balance = false
28.                 END IF
29.             END IF
30.         END For
31.     END While
32. END
```

The above algorithm time complexity and space complexities will be optimal if provided with the adjacency list of each node. The algorithm checks for the Signed graph (Sigraph) balance as follows:

A graph traversing should be started with an arbitrary vertex. From the given set of vertices V(G), let N be the arbitrary vertex we start the traversing. In line 2, vertex N will be set as *DISCOVERED* vertex. Line 3, we assume that the signed graph is balanced and initialize Balance as true. In line 4, arbitrarily the sign from sign function σ = {+, -} will be assigned to N. From Line 5 to 7, it will initialize the adjacency vertices of N as *UNDISCOVERED*. In line-8, as we use Depth-First search traversing technique in this algorithm. Depth-First search technique works using a Stack. We indicate the current position of the Stack we use a point ‘top’. When Stack is empty ‘top’ value is assigned to -1. From line 9 to 14, the signs of the adjacent vertices of N are been assigned using:

$$\sigma[p] = \sigma[N] * \sigma[(N, p)]$$

Once the adjacent vertex is assigned with a sign, it is added to the Stack and also marked as *DISCOVERED*. This procedure is repeated till all the vertices from the adjacency list of N are assigned with sign and marked as *DISCOVERED*. From line 15 to 32, we determine the balance of the given Signed Graph by exploring the adjacent vertices of N and their adjacent vertices. When Stack point ‘top’ is not equal to -1 (i.e. if stack point equal to -1 then stack is empty and algorithm terminates) and the Balance is *‘true’* then we take the vertex from the top of the stack say ‘p’ and delete the vertex ‘p’ from the stack. Mark the vertex ‘p’ as *DISCOVERED*. In line – 19 verify, are there any adjacency vertices of ‘p’. From here there are two possibilities to occur if the adjacency list is not empty:

(i) If the vertex ‘q’ (which is adjacent to ‘p’ vertex) is not *DISCOVERED*, then it is read as *DISCIVDRED* and Stack pointer (top) is incremented by 1. Vertex ‘q’ is pushed into stack and assigned with the sign using:

$$\sigma[q] = \sigma[p] * \sigma[(p, q)]$$

(ii) If the vertex ‘q’ is *DISCOVERED*, then based on the condition (iii) of Harary’s Structure Theorem the sign of ‘q’ and product of the sign of ‘p’ and the sign of the edge connecting (p, q) is compared. If the sign is same then the remaining vertices in stack are tested. If not same then the Balance is set to *‘false’* and the program terminates.

Before terminating the algorithm execution if the Balance is ‘*true’*, every vertex for the graph will traversed/

examined at most once. If the Balance is *'false'* then at which ever vertex condition (iii) of structure theorem doesn't hold good, algorithm will terminate and return the Balance variable value as *'false'*. This indicates that the given Signed Graph is not balanced.

The time complexity if this algorithm is as follows: Every vertex of the graph will be traversed at most once. For every vertex the run time is $\Theta(|V|)$. In Depth-First Search technique execution, every edge is also traversed at most once. Hence the time for traversing each edge is $\Theta(|E|)$. Therefore, the total time complexity is $\Theta(|V + E|)$.

The space complexity for this algorithm is the major advantage. Since we used Depth-First Search technique, each vertex is visited at most once. Since, the *DISCOVERED* vertices are removed from the Stack, maximum size of Stack at any point of time during the algorithm execution is $(V + 1)/2$. The Space complexity of this proposed algorithm is $\Theta((V + 1)/2)$.

3. Conclusion

The algorithm presented in this paper is based on Depth-First Search technique and it provides much better results in terms of space complexity than that of Breadth-First Search technique. This algorithm is very useful is the analysis of social networks for their balance and behavior. The structural balance of social network can be detected much easier using the signed graph balance. There are numerous applications of this algorithm like resources allocation in process execution without entering the process into deadlock state.

4. References

[1] E. Loukakis, "A Dynamic Programming Algorithm to test a signed Graph for Balance", International Journal of Computer Maths, Vol 80[4], Pg. No: 499 – 507, 2003.

[2] Harary F - "Graph Theory", Addison Wesley, 1969.

[3] Harary F, "On the Notion of Balance of a Signed Graph", Michigan Mathematical Journal, 2, Pg. No: 143 – 146, 1953 - 1954.

[4] Heider F, "Attitude and Cognitive Organization", Journal of Psych., 21, Pg. No: 107 – 112, 1946.

[5] Giuseppe Facchetti, G. Iacono, C Altafini, "Computing Global Structural Balance in large-scale Signed Social Networks", PNAS, vol. 108, N0. 52, Pg.No: 20953 – 20958, 2011.

[6] Arvind Srinivasan, "Local Balancing influences global structure in Social Networks", PNAS, Vol. 108, No. 5, Pg. No: 1751 – 1752, 2011.

[7] T. H. Cormen, C. E. Leisersom, R. L. Riveso, C. Stein, "Introduction to Algorithms", Prentice Hall of India, 2nd Edition, Pg. No: 540 -548, 2003.

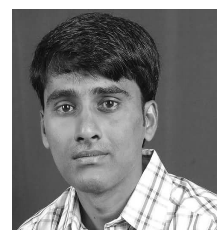
[8] Shi C J and Brzozowski, J A, "A Characterization of Signed Hypergraphs and its Applications to VLSI via Minimization and Logic Synthesis", Discrete Applied Mathematics, 90, Pg. No: 223,243, 1999.

[9] Zaslavsky T, "Charecterization of Signed Graph", Journal of Graph Theory, 5, Pg. No: 401 – 406, 1981.

[10] Knuth D, "Fundamental Algorithms", Vol. I of the Art of Computer Programming, MA: Addision – Wesley, 2nd Edition, 1973.

Authors Profile:

Mr. Sashi Kanth Reddy Avula complete Bachelor of Science from Acharya Nagarjuna University, Guntur, Andhra Pradesh in the year 2006 and Master of Computer Applications from Bangalore University, Bangalore in the year 2009. He is pursuing his Ph. D degree from JAIN University, Bangalore. He is working as Assistant Professor in Department of Master of Computer Applications at Acharya Institute of Technology, Bangalore. He is Life member of ISTE since 2014. His main area of research is in Advance Algorithms, Mobile Ad-Hoc Networks and Discrete Mathematics.



Mr. P Siva Kota Reddy completed his Bachelor of Science from Sri Venkateshwara University, Tirupathi in the year 2003 and Master of Science from the University of Mysore, Mysore in 2005. He completed his Ph. D from the University of Mysore, Mysore in 2009. He also completed Post Doctorate from Berhampur University, Berhampur in the year 2015. He is working as Professor in Department of Mathematics at Siddaganag Institute of Technology, Tumkur. He is Life Member in Ramanujan Mathematical Society, Life Member in Academy of Discrete Mathematics and Applications, Life Member of Indian Mathematical Society, Life Member of Journal of Current Science and Member in International Association of Engineers. He has published more than 120 research papers in various International Journals indexed by Thomson Reuters, Taylor and Francis. Five students got Ph. D awards under his guidance and he is guiding another six students for Ph. D.

