Volume:04 Issue:01 Pages:1501-1502 (2012) ISSN : 0975-0290

Algorithm to Find Clique Graph

A. Ashok Kumar
Department of Computer, Science,Alagappa Government Arts College, Karaikudi – 630003, India.
Email: ashokjuno@rediffmail.com
S. Athisayanathan
Research Department of Mathematics, St.Xavier's College(Autonomous), Palayamkottai – 627002, India.
Email: athisayanathan@yahoo.co.in
and **A. Antonysamy**
Ananda College, Devakottai.
Email: fr_antonysamy@hotmail.com

ABSTRACT

Let $V = \{1, 2, 3, \ldots, n\}$ be the vertex set of a graph G, $\mathcal{P}(V)$ the powerset of V and $A \in \mathcal{P}(V)$. Then A can be represented as an ordered n-tuple $(x_1x_2x_3\ldots x_n)$ where $x_i = 1$ if $i \in A$, otherwise $x_i = 0$ ($1 \le i \le n$). This representation is called binary count (or *BC*) representation of a set A and denoted as $BC(A)$. Given a graph G of order n, every integer m in $S = \{0, 1, 2, \ldots, 2^n-1\}$ corresponds to a subset A of V and vice versa. In this paper we introduce and discuss a sequential algorithm to find the clique graph $K(G)$ of a graph G using the *BC* representation.

Key Words: **binary count, clique, clique graph, powerset**.

Date of Submission: January 07, 2012 Revised: May 04, 2012 Date of Acceptance: June 02, 2012

1.Introduction

By a graph $G = (V,E)$ we mean a finite undirected graph without loops or multiple edges. $|V|$ and $|E|$ denote the order and size of G respectively. We consider connected graphs with atleast two vertices. A graph G is complete if every pair of vertices in G are adjacent. A clique of a graph is a maximal complete subgraph. The clique graph $K(G)$ of a given graph G has the cliques of G as its vertices and two vertices of $K(G)$ are adjacent if the corresponding cliques intersect in G. For basic concepts we refer [2,3].

In [1] Ashok, Athisayanathan and Antonysamy introduced a method to represent a subset of a set which is called binary count (or *BC*) representation. Given a graph G of order n, it is shown that every integer m in $S = \{0, 1, 2, \ldots, 2^n-1\}$ corresponds to a subset A of $V = \{1, 2, 3, \ldots, n\}$ and vice versa. Using this *BC* representation they introduced algorithms to find the subset A of the vertex set $V = \{1, 2, 3, \ldots, n\}$of a graph G that corresponds to an integer m in $S = \{0, 1, 2, \ldots, 2^n-1\}$, to verify whether A is a subset of any other subset B of V and also to verify whether the subgraph $< A >$ induced by the set A is a clique or not. Moreover a general algorithm is introduced to generate all cliques in a graph G using *BC* representation and proved the correctness of these algorithms and analyzed their time complexities.

In this paper we introduce algorithms to find the clique graph $K(G)$ of a graph G using *BC* representation.

2. Algorithms

In order to find the clique graph $K(G)$ of a given graph G, first we introduce an algorithm to check whether the two cliques of G have a common vertex in G.

Algorithm 2.1 Let G be a graph. Let $\zeta = \{C : C$ is a clique in $G\}$.

1. Let $C_1, C_2 \in \zeta$ in *BC* representation.
2. for $i = 1$ to n
3. if $C_1(i) = 1$ and $C_2(i) = 1$ then return *edge* $= (C_1, C_2)$
4. next i
5. return *edge* $= \Phi$
6. stop

Theorem 2.1 If C_1 and C_2 are two cliques in G, then Algorithm 2.1 finds whether C_1 and C_2 have a common vertex in G.

Proof: Let G be a graph of order n and $\zeta = \{C : C$ is a clique in $G\}$. Let $C_1 = (x_1, x_2, x_3, \ldots, x_n)$ and $C_2 = (y_1, y_2, y_3, \ldots, y_n)$ be their binary representation. We find a common vertex by scanning for any x_i, y_i such that $x_i = 1$ and $y_i = 1$ ($1 \le i \le n$). If $x_i = 1$ and $y_i = 1$ for some $i = 1, 2, \ldots, n$, then C_1 and C_2 have a common vertex. Thus this Algorithm 2.1 finds whether C_1 and C_2 have a common vertex or not in G.

Theorem 2.2 The time complexity of Algorithm 2.1 is $O(n)$ time.

Proof: In the worst case, the steps 2 to 4 are executed for n times, so that the time complexity of the Algorithm 2.1 is $O(n)$.

Example 2.1 For the graph G given in Figure 2.1, $V(G) = \{1, 2, 3, 4, 5, 6, 7\}$, $C_1 = \{1, 2, 3\} = (1110000)$, $C_2 = \{2, 4, 5\} = (0101100)$, $C_3 = \{2, 3, 5\} = (0110100)$, $C_4 = \{3, 5, 6\} = (0010110)$ and $C_5 = \{6, 7\} = (0000011)$. Now let us verify whether the cliques C_1 and C_4 of G have a common vertex. For $i = 1$ the values of $C_1(1) = 1$ and $C_4(1) \neq 1$, for $i = 2$ the values of $C_1(2) = 1$ and $C_4(2) \neq 1$, and for $i = 3$ the values of $C_1(3) = 1$ and $C_4(3) = 1$, therefore C_1 and C_4 will intersect in G so that there is an edge between the cliques C_1 and C_4 in $K(G)$. For the cliques C_2 and C_5, it is easy to check the Algorithm 2.1 returns no edge. So that C_2 and C_5 do not intresect in G.

Now we introduce an algorithm to find all the edges of a clique graph $K(G)$ of a graph G.

Algorithm 2.2 Let G be a graph of order n. Let $\zeta = \{C : C$ is a clique in $G\}$ and $|\zeta| = m$.

1. Let $\zeta = \{C_1, C_2, \ldots, C_m\}$.
2. $Cedge = \{\Phi\}$
3. for $i = 1$ to $m - 1$
4. for $j = i + 1$ to m
5. for (C_i, C_j) call Algorithm 2.1
6. if (C_i, C_j) is an edge, then $Cedge = Cedge \cup (C_i, C_j)$
7. next j
8. next i
9. stop

Theorem 2.3 Let $\zeta = \{C : C$ is a clique in $G\}$, the Algorithm 2.2 finds the edges of $K(G)$.

Proof: Let G be a graph of order n and $\zeta = \{C_1, C_2, \ldots, C_m\} \in G$. Cliques are represented in its *BC* form. If any two cliques has common vertices then the two cliques forms an edge of the clique graph $K(G)$. This can be obtained by $E(K(G)) = (C_i, C_j)$ if $Ci(k) = 1$ and $C_j(k) = 1$, for any $k = 1$ to n, Where $i = 1$ to $m - 1$ and $j = i + 1$ to m. Thus the Algorithm 2.2 finds the edges of $K(G)$.

Theorem 2.4 The edges of $K(G)$ can be found in $O(nm^2)$ time using the Algorithm 2.2

Proof: The steps from 3 to 7 are executed for m-1 times. Steps 4 to 6 are executed as inner loop for $(m - 1) + (m - 2) + \ldots + 1$ times. So the total time complexity for the steps from 3 to 7 is $O(m^2)$. Since the complexity of the Algorithm 2.1 is $O(n)$ by Theorem 2.2, the complexity of step 5 is $O(nm^2)$. Hence the time complexity of the Algorithm 2.2 is $O(nm^2)$.

Example 2.2 As in the Example 2.1, for the graph G given in Figure 2.1, $V = \{1, 2, 3, 4, 5, 6, 7\}$, $\zeta = \{C_1, C_2, C_3, C_4, C_5\}$ the set of all cliques in G and $m = 5$. The Algorithm 2.2, initially the set $Cedge = \{\Phi\}$ and for $i = 1$ and $j = 2$ by calling the Algorithm 2.1, it returns the edge (C_1, C_2) and the set *Cedge* is updated with an edge $Cedge = \{(C_1, C_2)\}$. Similarly we find all the edges of the clique graph $K(G)$. Hence the Algorithm 2.2 returns (C_1,C_2), (C_1,C_3), (C_1,C_4), (C_2,C_3), (C_2,C_4), (C_3,C_4), (C_4,C_5) as the edges of $K(G)$.

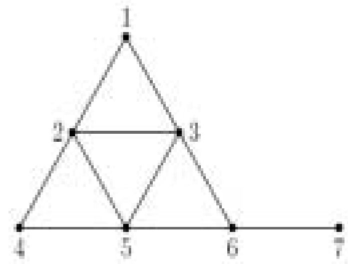

Figure 2.1: G

3. Conclusion

In this paper we have discussed sequential algorithms to check whether the two cliques of a connected graph G have a common vertex in G in $O(n)$ time and to find all the edges of the clique graph $K(G)$ of G in $O(nm^2)$ time. This clique graph algorithms can be used in cluster analysis.

References

[1]. A.Ashok Kumar, S.Athisayanathan, A.Antonysamy, "*Algorithm to Find All Cliques in a Graph*", International Journal of Advanced Networking and Applications, Vol. 2, No-2(2010), 597-601.

[2]. G. Chartrand and Ortrud R. Oellermann, "*Applied and Algorithmic Graph Theory*", McGraw-Hill International editions (1993).

[3]. K.R. Parthasarathy, "*Basic Graph Theory*", Tata McGraw-Hill Publishing Company Limited, New Delhi (1994)

Authors Biography

Dr.A.Ashok Kumar is working as Assistant Professor in Computer science at Alagappa Govt. Arts College, Karaikudi. He has 16 years of teaching experience in computer science.

Dr.S.Athisayanathan is working as Associate Professor in Research Department of Mathematics at St.Xavier's College(Autonomous) Palayamkottai. He has 30 years of teaching experience in Mathematics.

Dr.A.Antonysamy is working as Principal of Ananda College Devakottai, sivagangai dt. He has 37 years of teaching experience in Mathematics.