Sindh Univ. Res. Jour. (Sci. Ser.) Vol.45 (2) 393-398 (2013)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

A Weighted Edge Clique Problem

Z. U. A. KHUHRO[++], K. MEMON, F. N. MEMON, A. K. SOOMRO

Institute of Mathematics and Computer Science, University of Sindh, Jamshoro

Received 26th March 2013 and Revised 2nd June 2013

Abstract: This paper is an improvement to one of the most recent and an efficient algorithm Cliq that finds a clique from an undirected graph. The suggested improvements are detection of a clique from a random undirected graph where an addition of weight to its edges is given. In addition to that, the method is compared with the well-known existing methods of weighted clique problems.

Keywords: Maximum weighted clique problem, an efficient algorithm, DIMACS graphs.

1. INTRODUCTION

The maximum weighted clique problems are defined for finding a clique with the largest number of vertices in a given graph where the corresponding weight to the edges are assigned. Such a weight to the edges can be considered as a displacement vector between two vertices (Frank, 2013). There are some methods, which address the maximum weight clique problem by different ways such as MCS, BnB and ILS (Tomita, 2007; Tomita, 2010 and Andrade, 2012).

Carraghan and Pardalos in 1990 have presented a clique finding method that works for random graphs as well as for DIMACS graphs with up to 3000 nodes and with the density up to 25 percent (Sewell, 1998). In addition, a maximum weight clique discussed by Jain, 2011 for the graphs where the weights to graphs is given is the extension of the well-known Bron-Kerbosh method, 1973.

There is a well-known algorithm which enumerates all cliques in a graph is of Bron and Kerbosch (1973). The Kerbosch method uses branch and bound method, and finds all maximal cliques but it does not count the weight of the edges. The order of the Bron-Kerbosch method is $O(3^{n/3})$ (Jenelius, 2006, Christofides, 2012). Furthermore, the calculation of the upper bounds with branch and bound method discussed in Sewell, (1998).

The maximum weighted clique problem is NP hard so it is difficult to deal with such problems. Some methods spilt such graphs into smaller ones in order to reduce the computational burden (Liu, 2008, Frank, 2013).

Bioinforamtics also provides a huge storage capacity of networks, where the information on the molecular scale is encoded. DNA operations are used for solving travelling salesperson problem where DNA single strand are used for lengths on the edges with the corresponding DNA operations such as Merge, Annealing, Denaturation, Separation, and Discard (Kumlander, 2006; Wang, 2012). In Wang's paper, a DNA procedure is discussed for finding out the shortest path problems. It is very interesting problem which is defined for edge-weighted graph G = (V, E) which finds a path at starting and ending vertices such that the total weights on the path is smallest.

There are many applications of maximum weighted clique problems. Some of them are graph matching, computer vision, image matching, similarity searching and in structural biology, information retrieval (Babel, 1994; Brijnesh, 2004; Bahram, 2007; Batsyn, 2013).

This paper presents an efficient method that finds the maximum weight clique in a random undirected graph. The presented method will look for the edge weights in any graph. The method is extension of the method presented by Khuhro, *et al*., (2012)

In the result section, the comparison of the presented method with some well-known methods for DIMACS graphs is presented where the proposed method performs better. However, the DIMACS graphs are not weighted graphs but their density is applied to each edge in order to get the weighted graphs.

2. METHOD AND MATERIALS

Let us say G = (V, E) is a simple undirected graph with a set V of nodes and a set E of edges. Let $w: E \to R^+$ is a weighting function, which assigns to each edge i ε E a real number. For a clique C of G, the edge weight is defined as follows:

[++]Corresponding author, Z.U.A. Khuhro, email: kzainkhuhro@yahoo.com Ph. No: +92-333-1123976

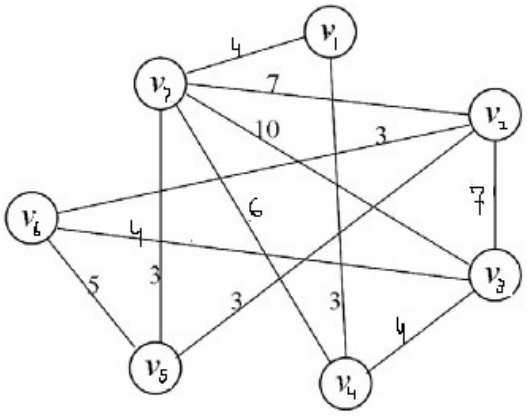
The first and second authors wish to be considered as the joint first authors of the article.

$$w(C) = \Sigma_{i \in C} w_i \quad (1)$$

Where w_i is the weight of the edges and a C is defined as follows:

$$C = \{v_j,\ j \in N\} \quad (2)$$

A adjacency matrix of a graph on n vertices is an n × n matrix $M(i, j)$ in which the entry $M(i, j)=1$ if there is an edge from vertex i to vertex j and is 0 if there is no edge from vertex i to vertex j.



(Fig-1): An example of a graph where seven nodes and their corresponding weights are given which is called a edge weight graph.

An adjacency matrix $M(i, j)$ of the given graph is formed by assigning 1 to a vertex where the two vertexes are connected and 0 where there is no connection between them. Matrix M is strictly upper triangular matrix because the graph is undirected graph. The strictly upper triangular matrix will help to reduce the number of iterations of the proposed method while searching for a clique.

The adjacency matrix of above graph having 7 nodes in strictly upper triangular matrix can be written as

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The purpose of this research is to find a maximum weight clique from an undirected graph (Fig-1). The method is extended to some well-known graphs called DIMCAS (William, 1995). The proposed method applied on benchmark DIMACS graphs, and the results are in Table-1. Computationally the presented algorithm performs well and is better for the maximum weight clique problems.

ALGORITHM:

The algorithm CLIQ (Khuhro, 2012) has been modified for weighted edges:

Function to find a clique (SIZE_G)

1: J=1
2: **While** (J<=SIZE_G)
3: TOP:=0
4: CLIQUE_SIZE:=0
5: MAX_SIZE:=0
6: TOP:=TOP+1
7: CLIQUE[TOP]:= n[i] an unprocessed node of N
8: CLIQUE_SIZE:=CLIQUE_SIZE+1
9: WEIGHTS[TOP]:=W[i]
10: **While** (CLIQUE≠Null)
11: TOP_NODE:=CLIQUE[TOP]
12: **If** (TOP_NODE has no unprocessed adjacent node)
13: **If** (CLIQUE_SIZE>MAX_SIZE)
14: MAX_CLIQUE:=CLIQUE
15: MAX_SIZE:=CLIQUE_SIZE
16: MAX_WEIGHTS:=WEIGHTS
17: **End if**
18: **If**(CLIQUE_SIZE=MAX_SIZE) AND (∑WEIGHTS<∑MAX_WEIGHTS)
19: MAX_CLIQUE:=CLIQUE
20: MAX_SIZE:=CLIQUE_SIZE
21: MAX_WEIGHTS:=WEIGHTS
22: **End if**
23: TOP:=TOP-1
24: **Else**
25: TOP:=TOP+1
26: CLIQUE[TOP]:=n[i] an unprocessed adjacent node of TOP_NODE that is also adjacent to all nodes in CLIQUE
27: CLIQUE_SIZE:=CLIQUE_SIZE+1
28: WEIGHTS[TOP]:=W[i]
29: **End of if**
30: **End of inner while**
31: **End of outer while**
32: **Return**

The proposed algorithm's description is as follows:

- N- Set of all nodes of a graph G
- CLIQUE - growing clique
- MAX-CLIQUE- nodes of maximum clique

- W-weights of edges in G
- SIZE_G-number of nodes in G
- CLIQUE_SIZE-number of nodes in clique
- MAX_SIZE- number of nodes in max_clique
- WEIGHTS-weight of an edge containing nodes in clique
- MAX_WEIGHTS-number of edges containing nodes in max_clique

3. RESULTS AND DISCUSSION

The proposed method is used to solve the undirected graphs having weights on the edges. The method tested on the benchmark graphs known as DIMACS graphs (William, 1995). However, the DIMACS graphs are not weighted graphs but their density is used as a weight for each edge. The selected problems having edge weights, the corresponding clique size and the edge size are defined in the Table-1. The edge weight is nothing but the edge density defined for each graph.

Table-1. The table shows the result of comparison of five algorithms Balas, 1996; Wood, 1997; Sewell, 1998; Ostergard, 2002; proposed method, for DIMACS graphs. Column one represents the names of DIMACS graphs. Second column is for the density, third column gives the number of nodes in graphs, whereas fourth column describes the size of clique is each graph. Fifth column shows the edges. Whereas 6-10 columns illustrate the ratio of time in seconds and clique for each method represented by A, B, C, D and E. This ratio is multiple of 10^5.

BENCHMARKS RESULTS OF DIMACS GRAPHS									
DIMACS Graphs	d	Nodes	Size	Edges	A	B	C	D	E
c-fat200-1	0.08	200	12	1534	0.03	0.02	-	0.01	0.00333
c-fat200-2	0.16	200	24	3235	0.02	0.02	-	0.01	0.00166
c-fat200-5	0.43	200	58	8473	0.13	0.13	-	7.81	0.00413
c-fat500-1	0.04	500	14	4459	0.08	0.02	-	0.07	0.00285
c-fat500-10	0.37	500	126	46627	0.64	0.03	-	0.07	0.03047
c-fat500-2	0.07	500	26	9139	0.11	0.08	-	0.08	0.00615
c-fat500-5	0.19	500	64	23191	0.24	0.16	-	10,442.64	0.01000
johnson16-2-4	0.76	120	8	5460	0.01	195.80	-	0.27	0.00500
johnson8-2-4	0.56	28	4	210	0.01	0.02	-	0.01	0.00000
johnson8-4-4	0.77	70	14	1855	0.01	0.18	-	0.01	0.00285
keller4.clq	0.65	171	11	9435	3.39	18.45	1.95	0.50	0.00888
hamming10-2	0.99	1024	512	518656	30.89	79.15	4.58	0.86	0.11507
hamming6-2	0.90	64	32	1824	0.01	0.01	-	0.01	0.00000
hamming8-2	0.97	256	128	31616	0.34	0.01	-	0.04	0.01812
hamming8-4	0.64	256	16	20864	1.74	79.15	4.58	0.86	0.01500
p_hat300-1	0.25	300	8	10933	0.80	1.47	2.00	0.04	0.00042
p_hat300-2	0.49	300	25	21928	5.83	10.05	2.79	0.99	0.00035
p_hat500-2	0.50	500	36	62946	228.11	267.10	-	428.80	0.00108
p_hat700-1	0.25	700	11	60999	16.08	40.32	12.53	0.67	0.00176
san200_0.7_2	0.70	200	18	13930	4.91	0.47	-	0.04	0.02400
san200_0.9_1	0.90	200	70	17910	0.26	11.48	-	0.27	0.01309
san200_0.9_2	0.90	200	60	17910	12.27	1052.00	-	4.28	0.01846
Brock200_1	0.75	200	21	14834	172.19	805.20	-	54.29	0.01882
Brock200_2	0.50	200	12	9876	1.90	3.83	1.58	0.05	0.01200
Brock200_3	0.61	200	15	12048	7.96	38.50	-	0.44	0.01666
Brock200_4	0.66	200	17	13089	26.00	92.95	4.16	0.98	0.01714
MANN_a9	0.93	45	16	918	0.02	0.10	-	0.01	0.00000

- A (Balas, 1996)-DEC alpha 300-400 AXP.
- C (Wood, 1997)-Sun Sparcstation 10
- B (Sewell, 1998)-
- D (Ostergard, 2002)-500MHz PC, Linux Operating System
- E (proposed method)-Linux Operating System, 500MHz.

DIMACS graphs whereas second column gives the density value and third column defines nodes. The fourth and fifth columns define the size of the clique and number of edges in the graph respectively. The next five columns give the computational times required for the five methods that are being considered. The times (seconds) given for the each method is a multiple of ten power five.

The selected methods presented in Ostergard 2002 show that the Ostergard method performs better at some instances such as the problems of type brock, keller and p_hat. This is due to some properties of the DIMACS graphs (Ostergard, 2002). But for some other instances such as type hamming and c-fat, the Balas, 1996 and Sewell, 1998 algorithms perform better.

In Wood 1997, some instances of types c-fat, Johnson, san200 and MANN have hidden clique that is much larger than what could be expected. It shows that the algorithm is very sensitive for some instances. The purpose to show the column C is that there are some instances where the method does not result in expected time.

Results of the proposed method are listed in final column of Table-1. Results for each DIMACS graphs are better as compared with the other four methods. For each graph the calculated time as compared to other methods is very low. The method is better for all the categories that are explained in Table-1, i.e. c-fat, brock, johson, p_hat, san, hamming, mann and keller that can be seen from Table-1. Better results have been achieved using the proposed method for all the DIMACS graphs but it is always hard to state for any algorithm that it is better than the others (Rozycki, 2010).

4. CONCLUSION

In this paper, the CLIQ method is modified in order to find the maximum weight clique. The weight can be assigned to vertexes and edges but the focus in this research is on the maximum edge weight clique problem (Fig-1). The method, when compared with some well-known existing methods performed better. All the comparison results are summarised in Table-1. However, the method performed well on DIMACS graphs but there are many other parameters in saying that the method is better than the other existing methods. It is therefore, can be concluded that the method is an efficient and finds the required maximum number of edge weight cliques in a small interval of time. The proposed method can also be applied on random graphs as well.

The future work will be to consider the vertex colouring and vertex weight of the graphs when looking for the maximum weighted cliques.

REFERENCES:

Andrade, D. V., M. G. C. Resende, and R. F. Werneck, (2012) Fast local search for the maximum independent set problem, *Journal of Heuristics,* 18 (1), 525-547.

Babel, L., (1994) A fast algorithm for the maximum weight clique problem, *Computing,* **52** (1), 31-38.

Bahram, A., G. Fred, K. Gary, and W. Haibo, (2007) Solving the maximum edge weight clique problem via unconstrained quadratic programming, *European journal of operational research, Elsevier,* **181** (2), 592-597.

Balas, E., and X. Jue, (1996) Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring, *Algorithmica, Springer*, **15**(5), 397-412.

Batsyn, M., B. Goldengorin, E. Maslov, and P. M. Pardalos, (2013) Improvements to MCS algorithm for the maximum clique problem, *Journal of Combinatorial Optimization, Springer,* 1-20. DOI 10.1007/s10878-012-9592-6.

Bron, C. and J. Kerbosch, (1973) Algorithm 457: finding all cliques of an undirected graph, *Communications of the ACM, ACM*, **16**(9): 575-577.

Brijnesh, J. J, F. Wysotzki, (2004) The maximum weighted clique problem and Hopfield networks, *Proc. of the 12th European Symposium on Artificial Neural Networks, ESANN2004* **, (4):** 331-336 **.**

Carraghan, R., and P. M. Pardalos, (1990) An exact algorithm for the maximum clique problem, *Operations Research Letters, Elsevier*, **9**(6), 375-382.

Christofides, D., K. Edwards, and A. D. King, (2012) A note on Hitting maximum and maximal cliques with a stable set, *Journal of Graph Theory,* **73**(3): 354-360.

Frank, M. R., and J. H. Dinitz (2013) On Maximal Partial Costas Latin Squares, *http://www.emba.uvm.edu/~jdinitz/preprints/costasPaper.jcmcc.pdf*

Jain, B. J., and O. Klaus, (2011) Extending Bron Kerbosch for Solving the Maximum Weight Clique Problem, *Data Structures and Algorithms, arXiv preprint arXiv: 1101.1266,* "http://arxiv.org/pdf/1101.1266v1.pdf", 1-10, 2011.

Jenelius, E., P. Tom and M. Lars-Goran, (2006) Importance and exposure in road network vulnerability analysis, *Transportation Research Part A: Policy and Practice, Elsevier*, **(40):** (7), 537-560.

Khuhro, Z. U. A., F. N. Memon, M. U. R. Maree and A. P. Harrison, (2012) Cliq: A clique finding algorithm, *Sindh Univ. Res. Jour.(Sci. Ser,)* **(44):** (2), 313-318.

Kumlander, D., (2006) Improving the maximum-weight clique algorithm for the dense graphs, *Proceedings of the 10th WSEAS International Conference on COMPUTERS Vouliagmeni, Athens, Greece, July 10-12, 2006*, 938-943.

Liu, F., K. Anatoliy, T. I. Axenovich, C. M. V. Duijn, and Y. S. Aulchenko, (2008) An approach for cutting large and complex pedigrees for linkage analysis, *European Journal of Human Genetics, Nature Publishing Group*, **(16):** (7), 854-860.

Ostergard, P. R. J., (2002) A fast algorithm for the maximum clique problem, *Discrete Applied Mathematics*, **(120):** 97–207.

Rozycki, P., and A. Jajszcyk, (2010) The weighted graphs approach for the GMPLS network reliability enhancement, *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE*, 608-614.

Sewell. E. C., (1998) A branch and bound algorithm for the stability number of sparse graph, *INFORMS, Journal on Computing*, **(10):** 438-447.

Tomita, E., and T. Kameda, (2007) An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments, *Journal of Global Optimization, Springer*, **(37):** (1), 95-111.

Tomita, E., S. Yoichi, H. Takanori, T. Shinya, and W. Mitsuo, (2010) A simple and faster branch-and-bound algorithm for finding a maximum clique, *WALCOM: Algorithms and computation, Springer Berlin Heidelberg*, 5942:191-203.

Wang, Z., Y. Zhang, W. Zhou, and L. Haifeng, (2012) Solving traveling salesman problem in the Adleman--Lipton model, *Applied Mathematics and Computation, Elsevie*, **(219):**, 2267-2270.

William, C., (1995) Combinatorial optimization: papers from the DIMACS Special Year, Combinatorial Optimization, *American Mathematical Society*, 20.

Wood, D. R., (1997) An algorithm for finding a maximum clique in a graph, *Operations Research Letters, Elsevier*, **(21):** 5, 211-217.