Modeling and Simulation of Quantum Memory Management Problem with Impetus of Graph Isomorphism, Graph Decyclization and Graph Pebbling

Jitendra Binwal[1], Aakanksha Baber[2]
[1]Professor and Head, [2]Research Scholar, Department of Mathematics
School of Liberal Arts and Sciences (SLAS)
Mody University of Science and Technology, Lakshmangarh-332311, Raj., India.
([*]dr.jitendrabinwaldkm@gmail.com, [**]baberaakanksha8@gmail.com)

Abstract
The cover pebbling number of graph is the minimum number of pebbles can be configurated in a manner where after a series of pebbling steps, it is possible to have at least one pebble on every vertex ofgraph. In case of directed trees, cover pebbling number is applicable insists of pebbling number. We apply the graph invariant of isomorphism in terms of decyclization and pebbling in quantum memory management problem. Any two isomorphic graphs whose parameters are equivalent to each other such as decyclization and pebbling numbers with impetus of quantum memory management problem.

Keywords: *CoverPebbling Number, Graph Isomorphism, Decyclization, Quantum Memory, Directed Acyclic Graph.*

AMS[2010]: *68R10, 94C15, 97K30, 05C30*

1. INTRODUCTION

Graph theory is the extension of new possibilities in several fileds. To explore the reason behind the main problem in easy manner, in a manner of way where everybody can easily understand [4]. The graph isomorphism demonstrate the similarity between the any two objects in real life, that is the graphs represent any two objects which are similar or differ to each other with the help of numbers of invariant [8]. The decyclization can be applied to analysis of Petri nets, evaluation of circuits and programs etc. The known methods are reviewed but they are not effective for big graphs [9]. Acyclic graphs are widely used in applications for example, in logical design algorithms of circuit design.

Graph pebbling can also be used as a model for the distribution of the sources of information over a network. Graph pebbling is a mathematical game and area of interest played on a graph with pebbles on the vertices [5, 2, 7]. When two graphs are isomorphic then their set of vertices, edges are same and degree of each vertices $v_i$are same [4, 5]. Now, consider a directed graph $G(V, A, W)$ with non-empty sets of vertices set V, arc set A and weighted set W. The decyclization of any directed graph is the removal of such edge or edges set which makes the graph acyclic i.e. minimum edges set removal from graph G [4, 9].

2. Graph pebbling and rubbling

Pebbling on graph is introduced by Chung. Consider, a graph$G$ contains set of vertices and set of edges where the incidence relationship is preserved. Pebbling step is the shifting of two pebbles from an arbitrary vertex and attachment of one pebble on its adjacent vertex. Graph pebbling is the collections of pebbling steps on the n number of vertices on graph with different configurations is known as the pebbling number of graphs G i.e. $\pi(G)$. The pebbling number of graphs is equal to number of vertices of graph is known as demonic. Consider $p(v)$ is the number of pebbles on vertex v

ISSN: 2233-7857 IJFGCN

[1, 2, 5].

Graph Rubbling is the supplement of graph pebbling. Often graph rubbling may be harder or easier than pebbling. On a graph, rubbling step is the shifting of one pebble from each adjacent vertex that is$s$ and v and attachment of one pebble on target vertex [7].The rubbling number of graphs G or $\rho(G)$ is the minimum number of pebbles on graph G which are randomly distributed all over the graph on vertices, after sequence of rubbling steps if target vertex is placed with one pebble from m number of pebbles.

2.1. The cover pebbling number $\boldsymbol{\gamma(G)}$

The cover pebbling number $\gamma(G)$of graph G is the minimum number of pebbles can be configurated in a manner where after a series of pebbling steps, it is possible to have at least one pebble on every vertex of graph [3, 7]. For an example,$\gamma(G) = 5$

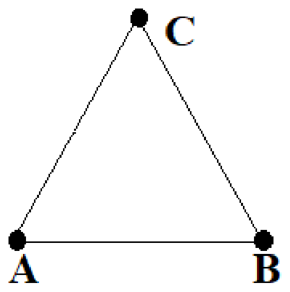


Figure 1. Cover pebbling step

The range for cover pebbling number, where n is the order of the graphs

$$2n - 1 \leq \gamma \leq 2^n - 1$$

Since the cover pebbling number can be depicted with pebbling number in some graphs as

$$\gamma(G) = 2\pi(G) - 1$$

The covering ratio is defined in terms of pebbling number, rubbling number and cover pebbling number of graph G.

$$\rho(G) = \frac{\gamma(G)}{\pi(G)}$$

2.2. $\boldsymbol{\pi(G), \gamma(G)}$ and $\boldsymbol{\rho(G)}$ in graph families

Graphs	$f(G)/\pi(G)$	$\gamma(G)$	$\rho(G)$
Complete graph	$\pi(K_n) = n$	$\gamma(K_n) = 2n - 1$	$\rho(K_n) = 2$
Path graph	$\pi(P_n) = 2^{n-1}$	$\gamma(P_n) = 2^n - 1$	$\rho(P_n) = 2^{n-1}$
Wheel graph	$\pi(W_n) = n$	$\gamma(W_n) = 4n - 9$	$\rho(W_n) = 4$

See [3, 7, 10]

2.3. Graph invariant of isomorphism in terms of decyclization and pebbling

Statement 1: The pebbling number and decyclization of any two isomorphic graphs are equivalent or similar to each other's. For an example, G_1 and G_2directed acyclic graphs in fig. 3 obtained from DTCPP [9] problem of fig. 2 are isomorphic to each other by the graph isomorphism definition.

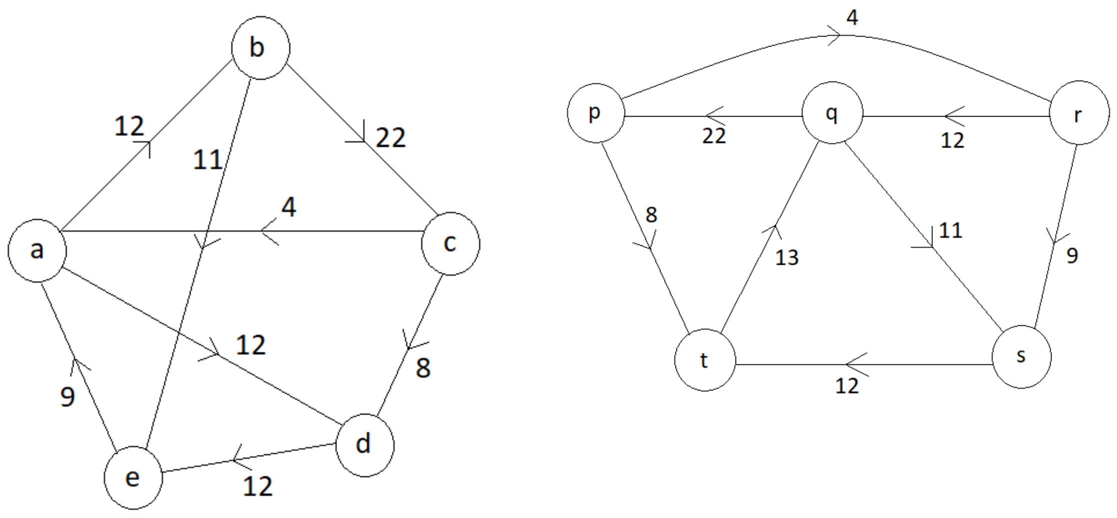


Figure 2. G and G' directed graphs

And their corresponding pebbling numbers are same that is $\pi(G_1) = \pi(G_2) = 5$, only if these two directed acyclic graphs are converted into undirected graphs.

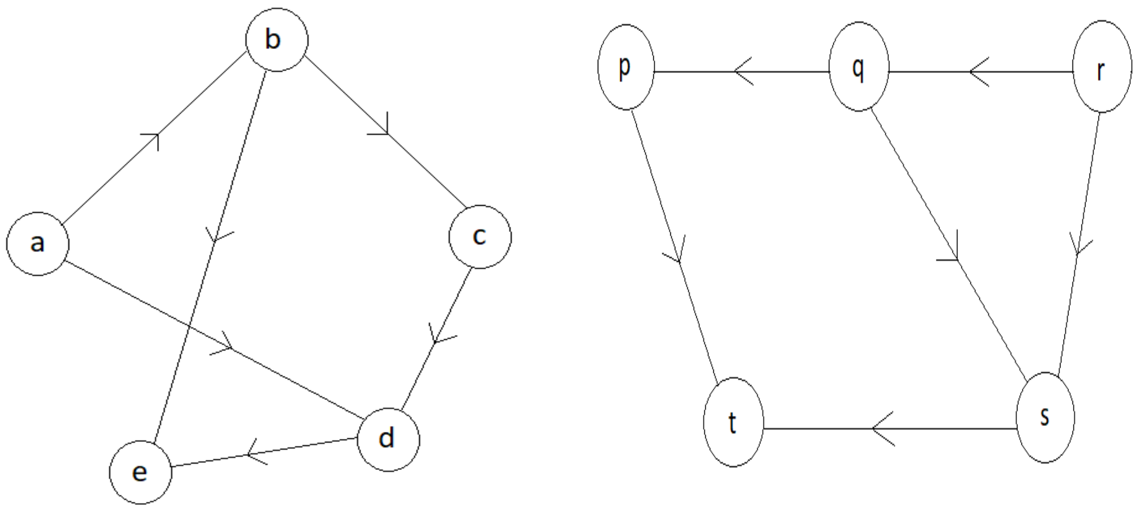


Figure 3.G_1 and G_2 directed acyclic graphs

The three aspects of graph theory (decyclization, isomorphism, pebbling number) are combined into one statement. After decyclization of two isomorphic graphs, these resulted graphs of G_1 and G_2 graphs are demonic graph and isomorphic to each other's.

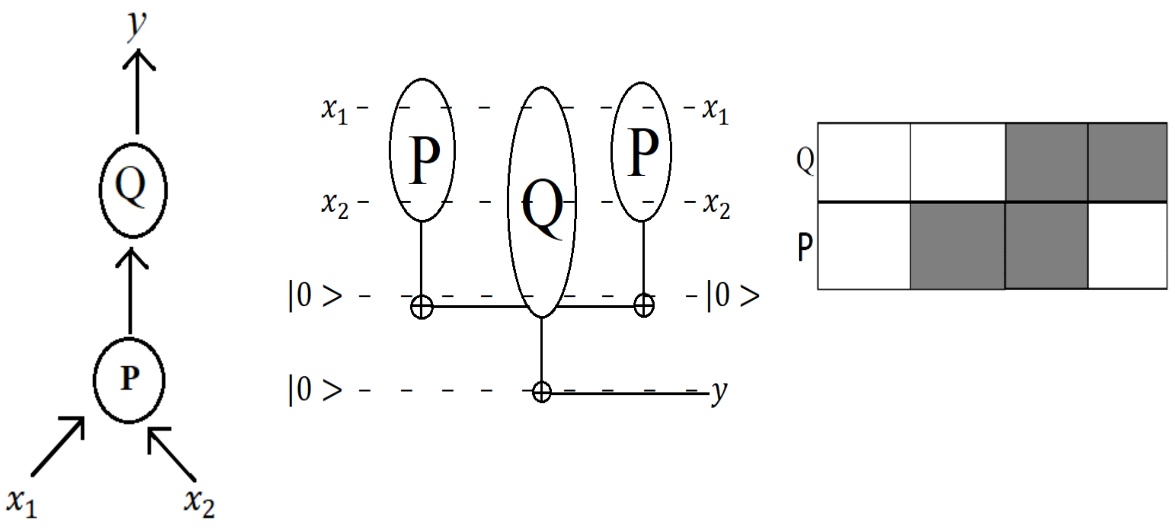
3. QUANTUM MEMORY PROBLEM

Quantum memory is an important component of quantum information processing applications such as quantum network, quantum repeater, linear optical quantum computation or long distance quantum communication.From a cybersecurity persuasion, the magic of qubits is that if a hacker tries to peak data'sduring transit, their delicate quantum states shatter. This means it is impossible for hackers to tamper with the network data without leaving a trace. Nowadays, many companies are taking advantage of this feature to create networks that transit it highly sensitive data. In theory, these networks are secure.

For quantum memory, quantum communication and cryptography are the future research areas.

Most important challenges is to create memories that can store the quantum information carried by light. The element ytterbium is a good insulator and protect quantum information even at the high frequencies that can be stored and quickly restored. Ytterbium is an ideal candidate for future quantum networks, since signals can't be copied or replicated. Since quantum memories can be made to travel farther and farther by capturing photons to synchronize them. In order to do this, this element becomes important to find the right materials for making quantum memories. There are several proofs for validation of reversible pebbling game on quantum memory problem such as applying the SAT (or Boolean satisfiability problem) problem to straight line programs or comparing with the existing approach [6].The quantum memory is the rendition of insignificant computer memory. Insignificant computers stores the information in terms of binary number that is 1's and 0's.

For quantum circuit's computation, large functions are decompose into smaller parts in order to deal with complexity. Since, each smaller part requires some resources in terms of qubits and quantum operations. The final quantum composed circuit does not evolve any of the intermediate values of large function (also stored in qubits). Whereas the quantum operations are reversible such that intermediate results can be uncomputed while computing the reverse order by performing the same operations [6]. Qubits which are initialized to$|0>$ is known as ancillae. For an example, For a DAG, $|x_1>|x_2>|0>\mapsto|x_1>|x_2>|y>$. Where$z_1=P(x_1,x_2)$ and$y=Q(z_1,x_1)$. While P and Q are two-input Boolean operation. Computation of quantum circuit designs can be represented using the DAGin fig. 2(a).



(a) (b) Bennett strategy (c)

Figure 4.

When y result is computed, all the intermediate valuez_1 are cleaned up. While a directed acyclic graph G is a sequence of reversible pebbling configuration $P=(P_1,\dots\dots,P_k)$ such that $P_1=\{0\}$ and $P_k=O$, where O is the all set of all sinks of directed acyclic graph G (or indegree).

For DAG of two node and three steps as in fig. 2(c), the complete sequence of pebbling configuration as$P_0=\{0\}, P_1=\{P\}$, $P_2=\{P,Q\}$, and $P_3=\{Q\}$.

The Bennett strategy leads to minimum number of gates and maximum number of qubits. While the pebbling strategy leads to maximum number of gates and minimum number of qubits.

4. Modeling and Simulating the Quantum memory management problem in graph invariant of isomorphism in terms of decyclization and pebbling

Here, we model the quantum memory management problem into a graph pebbling problem in such a way follows:

1) Computation problem of quantum memory is converted into a directed acyclic graph such as DAG (or Trees in graph theory).
2) Computation's first part is converted into each vertex such as $A = \{x_1, x_2\}$, then A is a vertex of directed acyclic graph that is each vertex/node is an operation.
3) Data dependency is converted as edges such as from vertex A to vertex B if B requires value of A.
4) Computed values as Qubits converted into pebbles such as x_i.
5) Quantum memory management games is represented as placing pebbles on graph vertices of DAG.
6) At last, if game is concluded then represents as DAG is pebbled.

We use the cover pebbling on the trees, since a trees and directed acyclic graphs consists sink and source. If a graph contains sink and source, it can't be pebbled with the pebbling step. Therefore, we use the cover pebbling in case of trees or DAG. Since outputs of the quantum memory computation game are pebbled via cover pebbling conditions and all other vertices are unpebbled. Here, it shows that this reversible pebbling game method of quantum memory management is capable of exploring the trade-off between space and time that is number of qubits and number of operations. We depict that even though the cover pebbling number of any two DAG's or trees are same, still they are not identical. For an example, two mobile phones with same features model of same company have outer layout similar to each other, but after being used by different persons the inner memory's are not identical or similar to each other of both mobile phones.

To be more specific, WhatsApp application is installed in every nine mobile phones out of ten nearby you. Even though the layouts, inputs and functioning features are same but each mobile phones contains different memory.Where the transformation of information or data's are represented as pebbles.

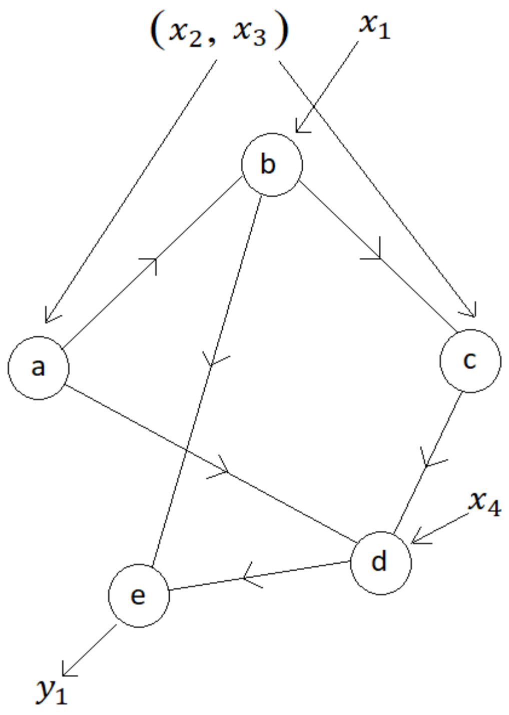
4.1. Reversible pebbling game of five nodes and nine steps

According to the quantum memory problem [6], we depicts the G_1 and G_2 DAG's in fig.1 as $|x_1 > |x_2 > |x_3 > |x_4 > |0 > \longmapsto |x_1 > |x_2 > |x_3 > |x_4 > |y_1 >$ for DAG G_1 and
$|x_1 > |x_2 > |x_3 > |0 > \longmapsto |x_1 > |x_2 > |x_3 > |y_2 >$ for DAG G_2.

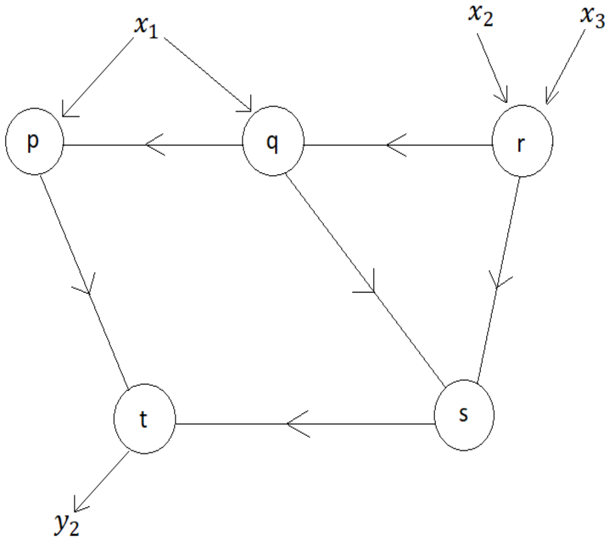
Where the intermediate results for DAG G_1,$z_1 = a(x_2, x_3)$, $z_2 = b(x_1, z_1)$, $z_3 = c(z_2, x_3)$, $z_4 = d(z_3, x_4)$ and$y_1 = e(z_2, z_4)$.

And for DAG G_2, $z_1^{'} = r(x_2, x_3)$, $z_2^{'} = q(x_1, z_1^{'})$, $z_3^{'} = s(z_1^{'}, z_2^{'})$, $z_4^{'} = p(x_1, z_2^{'})$ and$y_2 = t(z_4^{'}, z_3^{'})$. While, $a, b, c, d, e, p, q, r, s$ and t are two-input Boolean operations.

Computation of both quantum circuit design's can be represented using the directed acyclic graphsG_1 and G_2 in fig. 3.



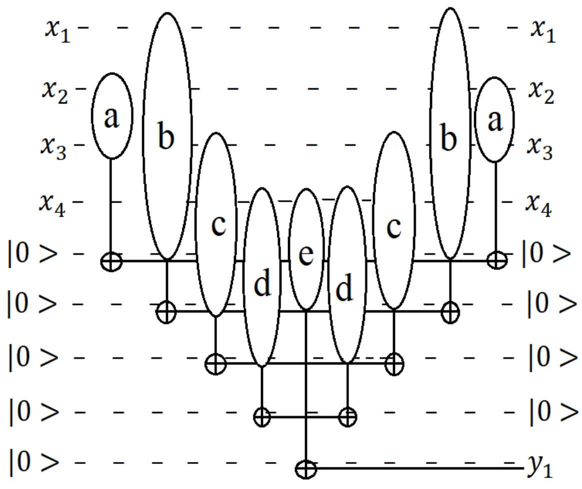
(a) G_1 DAG



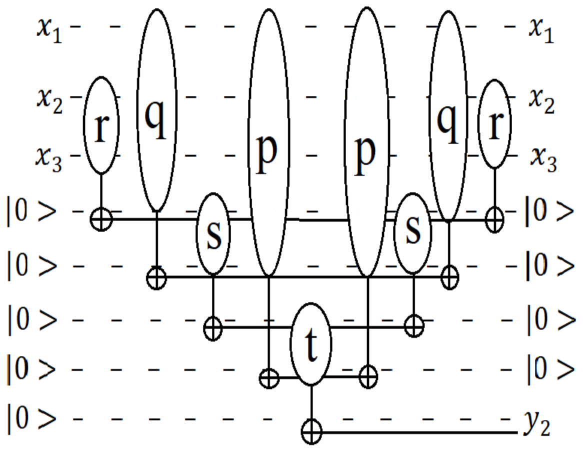
(b) G_2 DAG

Figure 5.

Once the y_1 and y_2 results of DAG's G_1 and G_2 are computed, all the intermediate values $z_1, z_2, z_3, z_4, z_1^{'}, z_2^{'}, z_3^{'}, z_4^{'}$ are cleaned up.



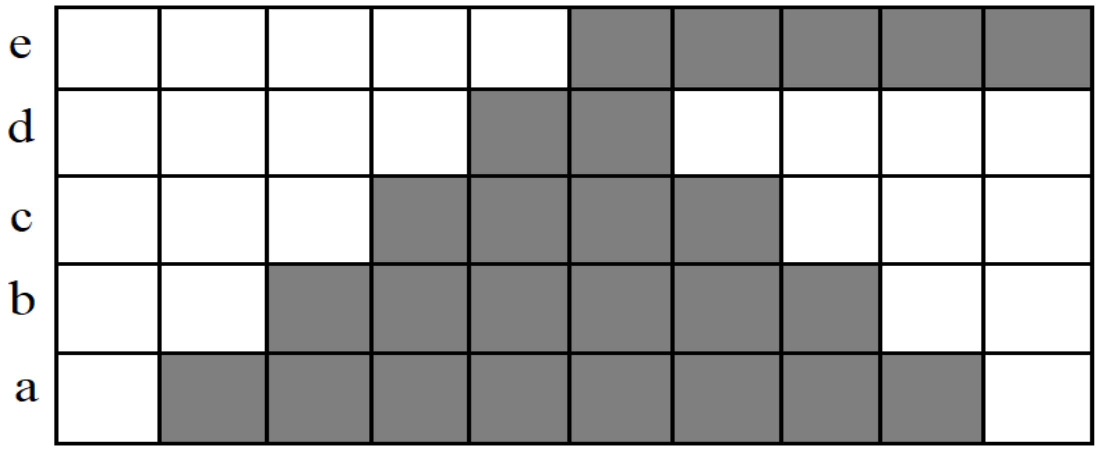
(a)Bennett strategy for G_1 DAG



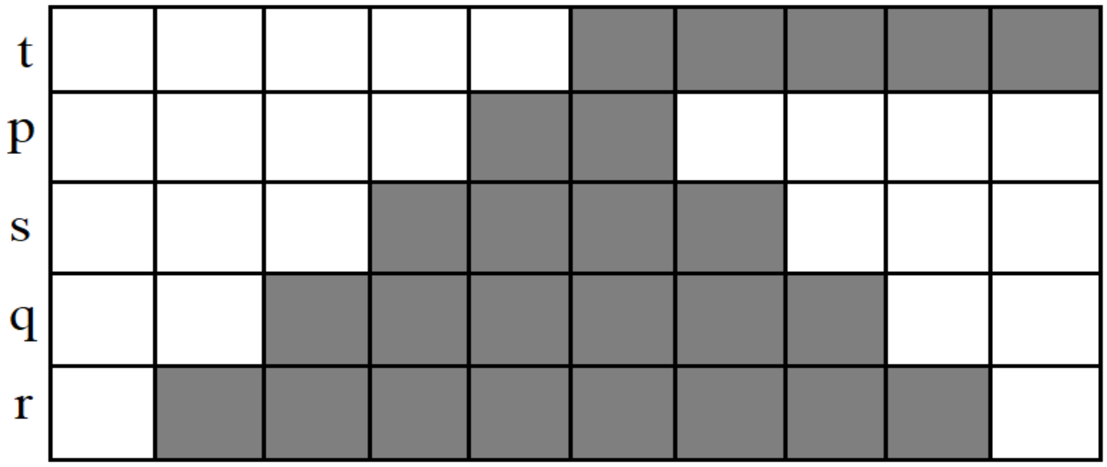
(b)Bennett strategy for G_2 DAG

Figure 6.

Bennett strategy results for G_1 and G_2 DAG's. since, they store the values for long time during which they are needed in fig. 6(a). While, pebbling strategy for G_1 and G_2 DAG's results as to make good usage of fewer memory. Though can't say pebbling method is better than Bennett method. Pebbling strategy optimize the space.The reversible pebbling game of G_1 and G_2 DAG's is a sequence of configuration of all pebbled vertices.



(a) Reversible pebbling strategy for G_1 DAG



(b) Reversible pebbling strategy for G_2 DAG

Figure 7.Reversible pebbling game of five nodes and nine steps

For G_1 DAG,

$$P_0 = \{0\}P_5 = \{a, b, c, d, e\}$$
$$P_1 = \{a\}P_6 = \{a, b, c, e\}$$
$$P_2 = \{a, b\}P_7 = \{a, b, e\}$$
$$P_3 = \{a, b, c\}P_8 = \{a, e\}$$
$$P_4 = \{a, b, c, d\}P_9 = \{e\}$$

For G_2 DAG,

$$P_0 = \{0\}P_5 = \{r, q, s, p, t\}$$
$$P_1 = \{r\}P_6 = \{r, q, s, t\}$$
$$P_2 = \{r, q\}P_7 = \{r, q, t\}$$
$$P_3 = \{r, q, s\}P_8 = \{r, t\}$$
$$P_4 = \{r, q, s, p\}P_9 = \{t\}$$

Since the pebbling numbers of G_1 and G_2 directed acyclic graphs in fig. 3 are not same that is $\pi(G_1) = 6$ and$\pi(G_2) = 5$. While their pebbling number are same if both DAG's are converted into undirected graphs that is $\pi(G_1) = \pi(G_2) = 5$.

Also, their corresponding cover pebbling numbers are same that is $\gamma(G_1) = \gamma(G_2) = 2(5) - 1 = 9$ for undirected graphs of fig. 3. But in case of directed acyclic graphs, their cover pebbling number is not same that is $\gamma(G_1) = 31$ and $\gamma(G_2) = 17$. More specifically, the cover pebbling number of any two directed acyclic graphs which are isomorphic in nature are not same.

With the help of SAT reversible pebbling game algorithm see [10], computations is carried out of quantum memory circuits. This algorithm is applied on the straight line program which results in pebbling the resulted DAG with different number of pebbles.

We have applied this algorithm on the graph invariant isomorphism in terms of decyclization and pebbling. At first, directed weighted graphs are decyclization into directed acyclic graphs or DAG's.

Since, the directed acyclic graphs are isomorphic in nature then their pebbling number are not same andtheir cover pebbling number of are alsonot same, then they are not identical during the storage time of intermediate values and after getting the output results that means y is pebbled and all intermediate values are cleaned up.

For an example, daily a larger scale of data's and information's are transferred on WhatsApp application. Even though the layouts, inputs and functioning features are same but each mobile phones contains different memory. Since the information's or data's are represented as pebbles. Then, after the transit of a data such as an image of hundreds of megabytes, there are some automatically stored caches of that particular image. Even though the two images on mobile phones are exact similar to each other's with same memory (let's say 315 MB), both the images will be stored with different caches.

If two exact mobile phones of model of same company clicks a same pictures, but their storage memory will be different.

5. CONCLUSION

We combine the three aspects of graph theory (decyclization, isomorphism, pebbling number) in one statement. We depict that the cover pebbling number of any two directed acyclic isomorphic graphs are not same; they are not identical which means their behaviour is different. We decrease the number of pebbles from DAG's so that the there is an increment in space of quantum memory. The complete sequence of pebbling configuration of both the DAG's during the computation are same or identical. The two DAG's are also isomorphic, their pebbling number and cover pebbling number are not same.

6. REFERENCES

1. Binwal J. et.al, Pebbling on Isomorphic Graphs, Indian Journal of Mathematics and Mathematical Sciences, 5(1) (2009), 17-20.
2. Chung F. R. K., Pebbling in Hyper cubes, SIAM J. Discrete Mathematics, 2 (1989), 467-472.
3. Crull B., Cundiff T., Feltman P., Hurlbert G. H., Pudwell L., Szaniszlo Z. and Tuza Z., The cover pebbling on graphs, Discrete Mathematics, 296 (2005), 15-23.
4. Deo N., Graph Theory with Applications to Engineering and Computer Science, PHI, (2001).
5. Gunda G., Pebbling on Directed Graphs, University of Dayton, (2004).
6. Meuli G.,Soeken M., Roetteler M., Bjorner N. and Micheli G. D., Reversible Pebbling Game for Quantum Memory Management, (2019).
7. Sieben N., Introduction to Graph Pebbling and Rubbling, Northern Arizona University, (2015).
8. Tiwari M., Binwal J., and Parihar C. L., Isomorphism of Undirected Plane Graphs using Graph Invariant: Probability Propagation Matrix, Journal of Indian Acad. Math., 29(1) (2007), 299-311.
9. Tiwari M., Binwal J., Parihar C. L., DTCPP- A heuristic program for testing decyclization in directed graphs and its isomorphic image by using combinatorial approach, Advances and Applications in Discrete Mathematics, 3(2), (2009).
10. Watson N. G. and Yerger C. R., Cover pebbling number and bounds for certain families of graphs, Bulletin of the Institute of Combinatorics and its Applications, 48 (2006), 53-62.