AN EXACT ALGORITHM FOR MINIMUM VERTEX COVER OF BLOOM INTERCONNECTION NETWORKS

Amutha. A

Assistant Professor, Department of Mathematics, The American College, Madurai.

Angel. D

Research Scholar, Department of Mathematics, Sathyabama University, Chennai.

Abstract: Bloom interconnection networks and covering sets are very important in computer architecture and communication techniques. In this paper, we solve the vertex cover problem for bloom interconnection networks. We also develop a linear time algorithm to find the exact solution of the vertex cover problem in bloom networks. Further we study the properties of bloom graphs and obtain its edge covering number.

Keywords: Bloom Networks, Minimum Vertex Cover, Minimum Edge Cover, Perfect Matching.

Introduction: combinatorial problems have become more important recently in the study of coverage, connectivity and fault tolerance in communication networks. Interconnection networks can be represented as an undirected graph, where a processor is represented as a vertex and a communication link between processors as an edge between corresponding vertices. The minimum vertex cover problem in interconnection network consists of finding the minimum number of processors which covers all the communication links.

The concept of vertex cover problem is motivated by the design of secure protocols for communication in interconnection networks. The vertex covering number of a graph is applied to measure the safety of a network. For instance, in a computer network, some servers play an essential role than others. Finding minimum vertex cover sets for that network whose vertices are the routing servers gives the optimal solution for designing the network defense strategy.

The minimum vertex cover problem (MVCP) is a classical optimization problem in computer science and is NP-Complete. Many of the problems like determining the number and location of radar installations, branch banks, shopping centers and waste disposal facilities can be formulated as the vertex cover problem. Covering sets have always been an attractive area of research due to its applicability in real world life, especially for service and emergency facilities [7]. The covering problem is also an area of strong concern in the design of both parallel structures and parallel algorithms. The minimum vertex covering sets for any structural model of parallel computation is both useful for the construction of efficient algorithms for that structure.

Most of the methods employed to solve MVCP are approximation algorithms or heuristics [3, 4, 5, 6, 8]. In contrast there has been surprisingly little work on covering problems in interconnection networks.
We first define several terms related to graphs. Let $G = (V, E)$ be an simple graph with vertex set V and edge set E. A graph G is connected if every pair of vertices u, v in $V(G)$ is joined by a path, otherwise, G is disconnected. We use the term graphs and networks interchangeably. The connectivity κ of a graph G is the minimum number of vertices whose removal results in a disconnected or trivial graph. This is also called vertex-connectivity. A subset S of V is called an independent set of G if no two vertices of S are adjacent in G. A clique of G is a subset S of V such that the subgraph of G induced by S is complete. An edge e of G is called a cut-edge or a bridge if removal of e disconnects the graph G. In other words, e is a bridge if and only if e is not contained in any cycle of G. If a graph G does not contain any bridges then G is called a bridgeless graph. A graph G is bipartite if and only if it does not contain any odd cycles. A connected graph is called Eulerian if and only if there are no odd vertices in G. A cycle passing through

all the vertices of a graph is called a Hamiltonian cycle. A graph containing a Hamiltonian cycle is called a Hamiltonian graph.

Bloom Graphs: Bloom interconnection networks are represented as undirected graphs whose vertices represent processors and edges represent inter processor communication links.

Bloom graphs, as mathematical structures, are interesting in and of themselves as they are both planar and regular which make them particularly attractive as potential structures for massively parallel computers. Motivated by the grid, cylinder and torus networks, Antony Xavier et al. in 2014, introduced the definition of bloom graph [1]. The bloom graphs are useful graph networks which can also be studied by specialists in dynamical systems and probability. They are very reliable networks as their vertex connectivity equals the degree of regularity.

The Bloom Graph denoted by $B_{m,n}$, where $m,n > 2$ is defined as follows: the vertex set is $V(G) = \{(x,y) \mid 0 \le x \le m-1;\ 0 \le y \le n-1\}$, two distinct vertices (x_1, y_1) and (x_2, y_2) being adjacent if and only if (i) $x_1 = x_2 - 1$ and $y_1 = y_2$ defines the vertical edges. (ii) $x_1 = x_2 = 0$ and $y_1 + 1 \equiv y_2\ (mod n)$ defines the horizontal edges in top most row. (iii) $x_1 = x_2 = m-1$ and $y_1 + 1 \equiv y_2\ (mod n)$ defines the horizontal edges in lower most row. (iv) $x_1 = x_2 - 1$ and $y_1 + 1 \equiv y_2\ (mod n)$ defines the slant edges [1]. For example, the grid view of bloom graphs $B_{4,\ 8}$ and $B_{3,\ 6}$ are shown in fig. 1(a) and 1(b) respectively and the flower view of bloom graph $B_{3,\ 6}$ and $B_{4,\ 6}$ is shown in fig. 2(a) and 2(b) respectively.

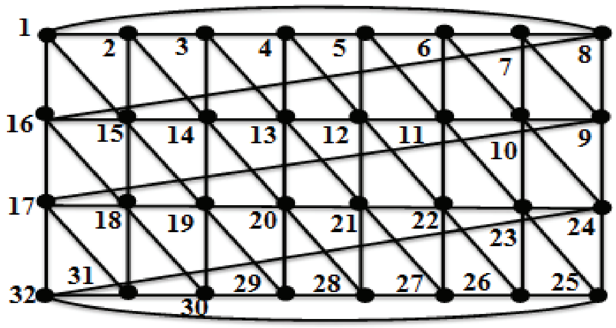

Figure 1(a). Grid view of $B_{4,\ 8}$

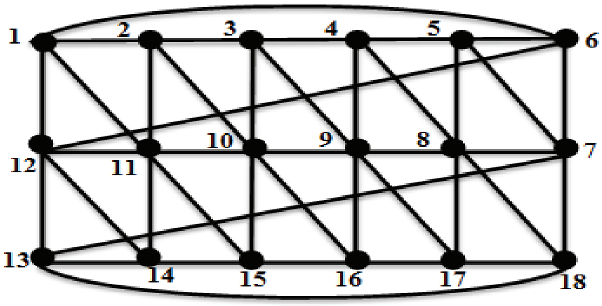

Figure 1(b). Grid view of $B_{3,\ 6}$

Antony Xavier et al. identified new topological representation for bloom graphs as grid view (fig.1(b)), and blooming flower view (fig.2(b)) and showed that these representations are isomorphic. Our algorithm of finding the minimum vertex cover set for bloom graphs works on the flower-like structure. To explain the flower structure of bloom networks, we consider $B_{4,6}$ an example. From fig.2(b), the inner most cycle which is at the center of $B_{4,\ 6}$ colored in green is a cycle of length 6 denoted by C_6. We call all those cliques of length 3 on top of C_6 colored pink, as petals. These n petals together with the center C_6 is called a floret (pink and green colored edges and the vertices on them) and is denoted by f_6 (see fig. 2(c)).

We denote by $L_{i,}$ the vertices in level i, where $1 \leq i \leq m + 1$.We explain the construction of the bloom graph using the flower view as follows.

Construction of $B_{m,n}$:

1. Input m, n.
2. Draw floret f_n.(fig. 3(a))
3. $i = 1$;
4. while ($i < m + 2$)do
5. Connect each L_{i+1} vertices on the petals by an edge such that it forms a cycle C_3 around the floret f_n.
6. Subdivide each of the edges in level L_{i+1} and call the new vertices and edges thus found by subdivision as L_{i+2} level vertices and L_{i+2} level edges.
7. $i = i + 1$
8. end while
9. stop.

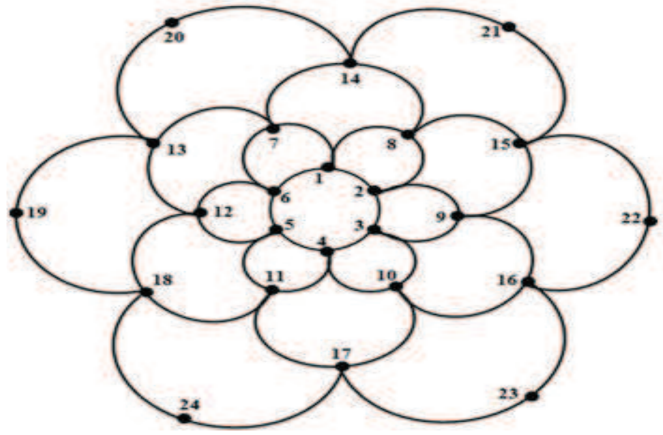

Figure 2(a). Flower view of $B_{3,6}$

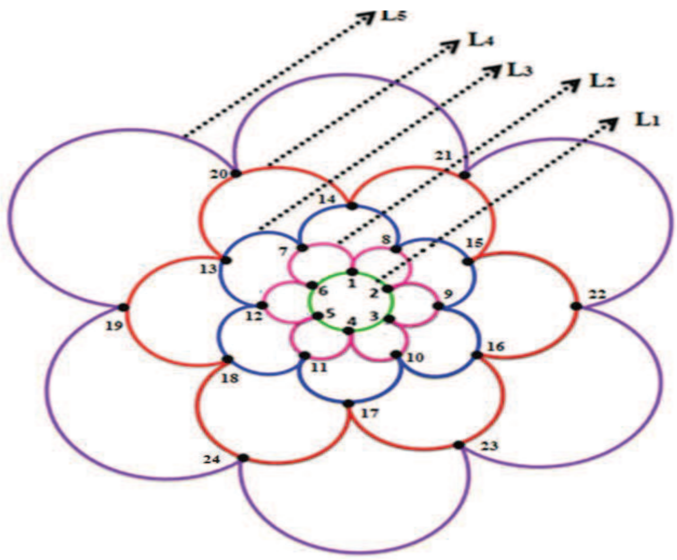

Figure 2(b). Flower view of $B_{4,6}$

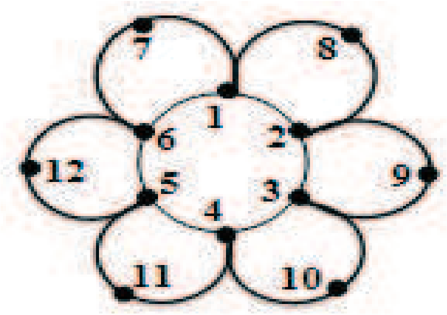

Figure 2(c). A Floret f_6

Minimum Vertex Cover: The minimum vertex cover consists in determining the minimum cardinality of a subset S of vertices such that all the edges are covered by those vertices in the set S. The number of elements in the smallest vertex cover set is called the covering number of G, denoted by β(G).We call this set as a β - set. For the graph in fig.3, β – set is given by S={ 2,5,8,11 }.

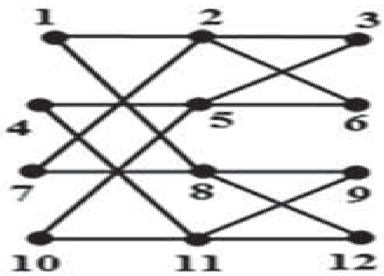

Figure 3. A graph with $\boldsymbol{\beta(G)} = 4$

We now give a linear time algorithm for finding the minimum vertex cover set of bloom graph. We denote by $L_{i,}$ the vertices in level i, where $1 \le i \le m + 2$. Let S_1 denote $\left\lceil \frac{n}{2} \right\rceil$ number of alternate vertices in the level L_{m+1} .

***Algorithm MVC-$B_{m,n}$*.**To find a minimum vertex cover set of a bloom graph.

Input : A bloom graph $B_{m,n}$ where $m > 1$, $n > 1$.
Output : A MVC set S of G.
Initialization: $S = \phi$; $i = 1$
If m is even then
while($i < m + 2$) do
$S = S \cup L_i$
$i = i + 2$
end while
end if
If m is odd then
while($i \le m$) do
$S = S \cup L_i$
$i = i + 2$
$S = S \cup S_1$
end while
endif
stop

The proof of correctness of the algorithm is given by the following theorem.

Theorem 1:

If $B_{m,n}$ is a Bloom graph then, $\beta(B_{m,n}) = \begin{cases} n\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{n}{2}\right\rceil, if\ m\ is\ odd. \\ n\left(\frac{m}{2}+1\right), if\ m\ is\ even. \end{cases}$

Proof: Let G be the bloom graph $B_{m,n}$ and let S be a minimum vertex covering set of G.
Case (i) : If m is odd. To cover vertices and edges on floret f_n we choose n vertices on C_n. That is, n number of L_1vertices are required to cover all the edges of f_n. Since all L_2 edges are already covered by L_1 vertices, we then choose all n number of L_3 vertices which covers all L_3 and L_4 edges. Proceeding in this way, up to $\left\lceil\frac{m}{2}\right\rceil$ times, we can have all edges except the level L_{m+2} edges are covered. Now, to cover L_{m+2} edges, we require $\left\lceil\frac{n}{2}\right\rceil$ vertices. So clearly the set S will contain the vertices on levels $L_1, L_3, L_5, \dots, L_{\lceil\frac{m}{2}\rceil}$ and $\left\lceil\frac{n}{2}\right\rceil$ number of L_{m+2} vertices. Adding all the vertices in S, we get the covering number as $n\left\lceil\frac{m}{2}\right\rceil + \left\lceil\frac{n}{2}\right\rceil$.
Case (ii) : If m is even. To cover vertices and edges on floret f_n we choose n vertices on C_n That is, n number of L_1 vertices, is required to cover all the edges of f_n. Since all L_2 edges are already covered by L_1 vertices, we then choose all n number of L_3 vertices which covers all L_3 and L_4 edges. Proceeding in this way, up to $\frac{m}{2}+1$ times, we can have all edges of the graph covered. Thus the set S will contain the vertices on levels $L_1, L_3, L_5, \dots, L_{m+1}$. Adding all the vertices in S, we get the covering number as $n\left(\frac{m}{2}+1\right)$.
Suppose if S is not minimum. Then there exists a covering set D which is minimum. If this is the case, then leaving out a single vertex in any level L_i from the set S will leave the edges uncovered and so S will not be a covering set. Therefore, S has to be the MVC set.

The vertices in a $\beta-$set can be used for various purposes, since every communication link will be under the coverage of one or more nodes. Router locations and backbone construction in wireless networks are designed based on the minimum vertex cover sets which can enhance the routing procedure.

Theorem 2:Bloom graphs are bridgeless.
Proof: Let G be a bloom graph. By the construction of these graphs we observe that, for all m and n, all edges in G lie on cycles. Hence we conclude that bloom graphs are bridgeless.

Theorem 3: Bloom graphs are Eulerian.
Proof: Let G be a bloom graph. These are4-regular graphs and hence there are no odd degree vertices in G. Therefore G is Eulerian.

Theorem 4:Bloom graphs are non-bipartite.
Proof: Let G be a bloom graph $B_{m,n}$. From the flower view of $B_{m,n}$, we see that all bloom graphs contains cycles C_3 which are odd cycles and hence non-bipartite.
Theorem 5: *[1]*Bloom graphs are Hamiltonian.

Invertible Graphs: In the design of network security, the primary focus is on defending the routes (edges) of the network by placing the minimum number of defenders in the network. However, in several practical applications, apart from defending the network, one has to consider the problem of replacing a defender by another when there is any defect in the network. To address this problem we studied the concept of invertible graphs. Invertible graphs are a subclass of bipartite graphs. Even though these are quite a small group of graphs they find wide applications where the independence and covering numbers play an important role, since these graphs are special in the sense that an independent set is also a covering set. Example of an invertible graph is shown in fig.4.

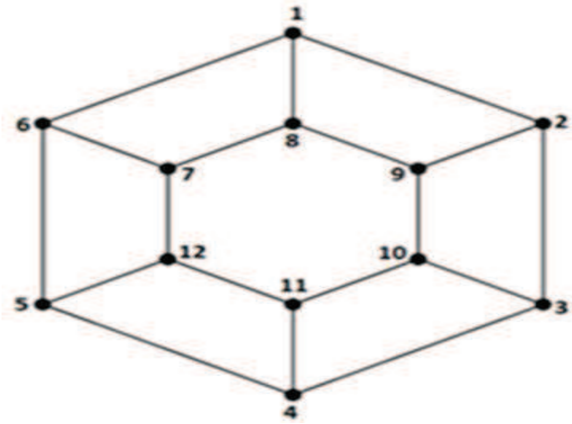

Figure 4. An invertible graph

Theorem 6: Every bloom graph is not invertible.
Proof: Every bloom graph contains at least n number of 3-cycles in it. Since odd cycles are not invertible, $B_{m,n}$ are not invertible graphs. A problem that might seem closely related to vertex cover is the edge cover problem. We also obtain the minimum edge covering number of the bloom interconnection networks.

Minimum Edge Cover And Perfect Matching: A set of edges of G is called an edge cover if that set covers all the vertices in G. The cardinality of the minimum edge cover set is called the edge covering number denoted by〚 β〛^' (G). Finding the minimum edge cover is called the edge covering problem.

Edge covers can be applied in network analysis. Another area where the edge covering number plays a role is the traffic phasing problem. Vertex cover and edge cover are closely related to perfect matching. Thereby, we were fascinated to find the kind of graphs for which, the edge covering and vertex covering numbers are equal.

A perfect matching M in G is a maximum number of non-adjacent edges with the property that every vertex is incident with an edge of the matching. M is always a minimum edge cover. For the graph in fig.5, the minimum edge cover set is given by $S = \{ (1,2), (4,5), (3,6), (7,8) \}$.

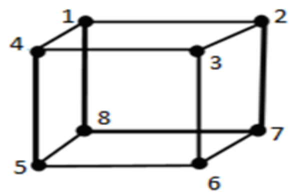

Figure 5. A graph with $\beta'(G) = 4$

Proposition 7: If $B_{m,n}$ is a Bloom graph with even number of vertices then G contains a perfect matching.
Proposition 8: If $B_{m,n}$ is a Bloom graph with even number of vertices then G contains a near perfect matching.

Theorem 9: Let $B_{m,n}$ be a bloom graph. Then, $\beta'(B_{m,n}) = \begin{cases} \frac{mn}{2}, if either\ m\ or\ n\ is\ even \\ \left\lceil \frac{mn}{2} \right\rceil, if\ m\ and\ n\ are\ odd \end{cases}$

Proof: Let $B_{m,n}$ be a bloom graph. Case (i) : If either m or n is even. Then mn is even. Since G contains even number of vertices, we conclude by proposition 7 that G has a perfect matching. This perfect matching is a minimum edge cover for G. Hence $\beta'(B_{m,n}) = \frac{mn}{2}$. Case (ii): If both m and n are odd. Then

mn is odd. Since G contains odd number of vertices, we conclude by proposition 8 that G has a near perfect matching. Therefore, $\beta'(B_{m,n}) = \frac{mn+1}{2} = \left\lceil\frac{mn}{2}\right\rceil$.

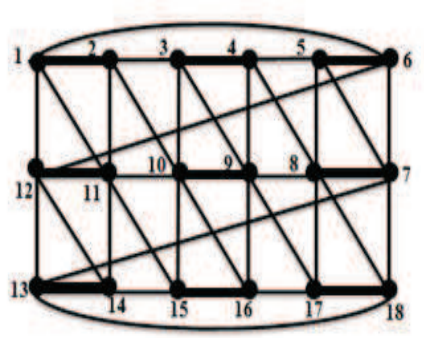

Figure 6. A perfect matching of $B_{3,\ 6}$

Conclusion: Construction and analysis of covering sets and invertible graphs helps us in the security of the networks as these sets are useful in monitoring the edges of a network.We have identified an algorithm which solves the vertex cover problem for bloom interconnection networks. Further, we have presented some of the properties of bloom graphs. It is interesting to observe that how our study for the vertex cover problem sets the stage for a much tighter analysis and we believe that this should be the general approach used when designing exact algorithms for NP- hard problems. Our methodology is useful to solve the covering problem for parallel architectures like pancake and shuffle exchange networks.

References:

1. Antony Xavier. D, Deeni C.J, Bloom Graph, International Journal of Computing Algorithm, vol. 3, 2014, pp. 521-523.
2. Bhat. P.G, Surekha R, Inverse Independence Number of a Graph, International Journal of Computer Applications, vol. 42, 2012, pp. 9-13.
3. Henning Fernau, Fedor V. Fomin, Gee varghese Philip, Saket Saurabh, On the parameterized complexity of vertex cover and edge cover with connectivity constraints, Theoretical Computer Science, vol. 565, 2 February 2015, pp. 1–15.
4. Jianhua Tu, A fixed-parameter algorithm for the vertex cover P3 problem, Information Processing Letters, vol. 115, Issue 2, February 2015, pp. 96–99.
5. Kartik Shah, Praveen kumar Reddy, R. Selvakumar, Vertex Cover ProblemRevised Approximation Algorithm, Artificial Intelligence and Evolutionary Algorithms in Engineering Systems Advances in Intelligent Systems and Computing, vol. 324, 2015, pp. 9-16.
6. Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseininia, Mark Goh, Covering problems in facility and location: A review, Computers and Industrial Engineering, vol. 62, pp. 368-407.
7. Vedat Kavalci, Aybars Ural, Orhan Dagdeviren, Distributed Vertex Cover Algorithms For Wireless Sensor Networks, International Journal of Computer Networks & Communications vol.6, No.1, 2014, pp. 95-110.
8. Wang Zhaocai, Huang Dongmei, Pei Renlin, Solving the Minimum Vertex Cover Problem with DNA Molecules in Adleman-Lipton Model, Journal of Computational and Theoretical Nanoscience, vol. 11, February 2014, pp. 521-523.
9. Yasuaki Kobayashi, Computing the path width of directed graphs with small vertex cover, Information Processing Letters, vol. 115, Issue 2, February 2015, pp. 310–312.

***