Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 57 (2015) 592 – 595

Procedia Computer Science

Improved Bounds on Forwarding Index of Networks

N. Parthiban[a], Indra Rajasingh[b,*], R. Sundara Rajan[b]

[a]School of Computing Science and Engineering, VIT University, Chennai, India
[b]School of Advanced Sciences, VIT University, Chennai, India

Abstract

In this paper we introduce a technique to compute an improved bound on edge-forwarding indices of graphs. Further we prove that the bound is sharp for cylinder, torus and certain trees.

Keywords: Edge-forwarding index, Cylinder, Torus, Trees

1. Introduction

There are a large number of graph optimization problems which arise in network design and analysis. A well-designed interconnection network makes efficient use of scarce communication resources and is used in systems ranging from large super computers to small embedded system on a chip. A designer of interconnection networks has to take into account communication speed, high robustness, rich structure, fault tolerance and fixed degree [1].

Routings are important functions of communication networks. The choice of a routing in a network directly affects efficiency of communication and performance of the network. There are many parameters to measure the quality of a routing. In this paper we consider one of the parameters namely the forwarding index, which is used to measure the load of a vertex or the congestion of an edge. It is quite natural that a good routing should not load any vertex or edge too much, in the sense that not too many paths specified by the routing should go through it.

Let G be a connected undirected graph or a strongly connected digraph with order n. A routing R in G defines a set of $n(n-1)$ fixed paths for all ordered pairs (x,y) of vertices of G. The path $R(x,y)$ specified by R carries the data transmitted from the source x to the destination y. If $R(x,y)$ is not a direct edge, then the internal vertices of $R(x,y)$ can serve as a forwarding function for the data being communicated between the vertices.

The congestion of an edge e in a routing R is the number of paths of R going through it, and is denoted by $\Pi(G,R,e)$. The edge-forwarding index of a routing R in a graph G is defined as

$$\Pi(G,R) = \max \Pi(G,R,e)$$

where the maximum is taken over all edges e of G. Then, the *edge-forwarding index* of G is defined as

$$\Pi(G) = \min \Pi(G,R)$$

where the minimum is taken over all routings R of G [4].

Forwarding indices for star graphs, cayley graphs, k-connected graphs, folded n-cubes and orbital regular graphs are found in [1-9]. Xu et. al. have given survey of forwarding index problems is found in [11].

Corresponding author.
E-mail address: parthiban24589@gmail.com

*This work is supported by Project No. SR/S4/MS: 846/13, Department of Science and Technology, SERB, Government of India.

Peer-review under responsibility of organizing committee of the 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
doi:10.1016/j.procs.2015.07.404

2. Main Results

A lower bound for $\Pi(G)$ available in the literature [4] is given in the following theorem 1.

Theorem 1 : For any connected graph $G = (V, E)$

$$\Pi(G) \geq \frac{1}{|E(G)|} \sum_{(u,v) \in V \times V} d(u, v).$$

The bound given in Theorem 1 involves distance between every pair of vertices in a graph G. The following simple observation is a major breakthrough in obtaining an improved bound for the edge forwarding index without using distance matrix.

For $S \subseteq E(G)$. Define $\Pi(G, R, S) = \cup_{e \in S} \Pi(G, R, e)$.

Lemma 2 : *Let S be an edge cut of G such that the removal of edges of S leaves G into 2 components G_1 and G_2.*

$$\text{Then } \Pi(G) \geq \frac{2|V(G_1)||V(G_2)|}{|S|}.$$

Proof. For a in G_1 and b in G_2, each of $R(a, b)$ and $R(b, a)$ passes through at least one edge of S. Therefore $\Pi(G, R, S) \geq 2|V(G_1)||V(G_2)|$. Hence $\Pi(G, R) \geq \frac{2|V(G_1)||V(G_2)|}{|S|}$, for all R. Thus $\Pi(G) \geq \frac{2|V(G_1)||V(G_2)|}{|S|}$.

Corollary 3 : *Let e be a cut edge of G such that removal of e leaves G into 2 components G_1 and G_2.*

$$\text{Then } \Pi(G) \geq 2|V(G_1)||V(G_2)|.$$

3. Forwarding Indices of Cylinder and Torus

We begin with cycle C_n on n vertices.

Algorithm A

Input :

Cycle C_n

Algorithm

1. Label the vertices of C_n with consecutive numbers $0, 1, \ldots, n-1$ in the clockwise sense.
2. Let $R(a, b)$ denote a shortest path between a and b. If there exist two shortest paths between a and b, choose the one in the clockwise direction.

Output : $\Pi(C_n) = \lfloor n/2 \rfloor \lceil n/2 \rceil$

Proof. $S = \{(0,1), (\lceil n/2 \rceil, \lceil n/2 + 1 \rceil)\}$ is an edge cut of cycle C_n and splits it into two components with $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ vertices. By Lemma 2, $\Pi(G) \geq \lfloor n/2 \rfloor \lceil n/2 \rceil$. By algorithm A, $\Pi(G) = \lfloor n/2 \rfloor \lceil n/2 \rceil$.

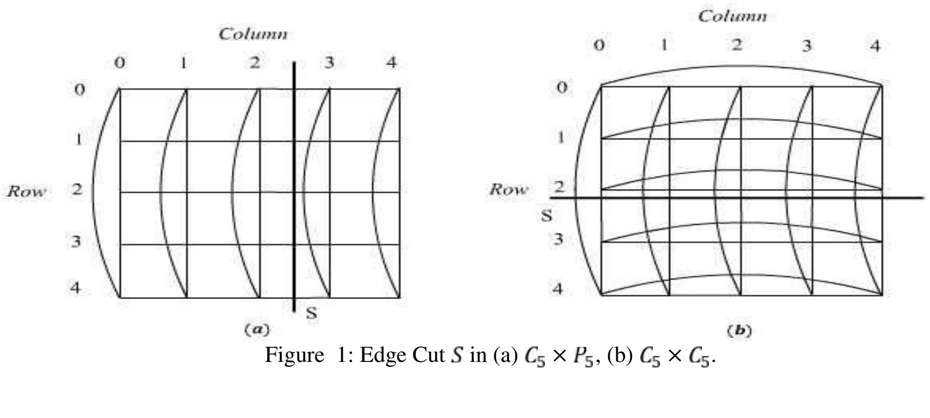

Figure 1: Edge Cut S in (a) $C_5 \times P_5$, (b) $C_5 \times C_5$.

Algorithm B

Input :

Cylinder $C_m \times P_n$/ Torus $C_m \times C_n$

Algorithm

1. Label the j^{th} row of $C_m \times P_n / C_m \times C_n$ as $n\ \ + 1, n\ \ + 2,...,n\ \ + n$ from left to right, where $0 \leq j < m$.
2. Let $R(a, b)$ denote a shortest path between a and b. If there exist more then one shortest path between a and b choose the one that moves from a^{th} row to b^{th} row in the clockwise direction.

Output :

$$\Pi(C_m \times P_n) = m\ \ \text{x}\{\frac{2 \times \lceil n/2 \rceil m \times (n\ \ \ - \lceil n/2 \rceil m)}{n}, \frac{\lceil m/2 \rceil n \times (n\ \ \ - \lceil m/2 \rceil n)}{m}\}.$$

$$\Pi(C_m \times C_n) = m\ \ \ \{\frac{\lceil n/2 \rceil m \times (n\ \ \ - \lceil n/2 \rceil m)}{n}, \frac{\lceil m/2 \rceil n \times (n\ \ \ - \lceil m/2 \rceil n)}{m}\}.$$

Proof. The set of edges S between $\lceil n/2 \rceil - 1^{th}$ column and $\lceil n/2 \rceil^{th}$ column is an edge cut of $C_m \times P_n$ and splits it into two components with $\lceil n/2 \rceil m$ and $(n\ \ \ - \lceil n/2 \rceil m)$ vertices respectively. By Lemma 2, $\Pi(C_m \times P_n) \geq \frac{2 \times \left|\frac{n}{2}\right| m \times (n\ \ -\left|\frac{n}{2}\right| m)}{n}$. By algorithm B, congestion on edges between $(i-1)^{th}$ and i^{th} column is $2 \times i\imath\ \ \times (n\ \ \ - i\imath\ )$ and congestion on each of the edges between $(i-1)^{th}$ and i^{th} column is $\frac{2 \times i\imath\ \times (n\ \ - i\imath\ )}{n}$. Congestion on edges between $(j-1)^{th}$ and j^{th} row is $2 \times j\imath\ \times (n\ \ \ - j\imath\ )$ and congestion on each of the edges between $(j-1)^{th}$ and j^{th} row is $\frac{j\ \times (n\ \ - j\ )}{m}$.

Remark : *Routing algorithms A and B give minimum congestion of any edge in* C_n, $C_m \times P_n$ *and* $C_m \times C_n$ *respectively.*

4. Forwarding Indices of Cylinder and Torus

In this section we compute the forwarding index for certain well known tree architectures. A tree is a connected graph that contains no cycles. Trees are the most fundamental graph-theoretic models used in many fields: data structure and analysis, design of algorithms, combinatorial optimization and design of networks [12].

The most common type of tree is the binary tree. A binary tree is said to be a complete binary tree if each internal node has exactly two descendents. These descendents are described as left and right children of the parent node. The complete binary tree of height r, denoted by T_r has exactly $2^r - 1$ vertices. The 1-rooted complete binary tree T_r^1 is obtained from a complete binary tree T_r by attaching to its root a pendant edge. The new vertex is called the root of T_r^1 and is considered to be at level 0. The k-rooted complete binary tree T_r^k is obtained by taking k vertex disjoint 1-rooted complete binary trees T_r^1 on 2^r vertices with roots say $r_1, r_2, \ldots, r_k$ and adding the edges (r_i, r_{i+1}), $1 \leq i \leq k-1$ [12].

Theorem 4 : *Let* G *be a complete binary tree* T_r. *Then* $II(G) = 2^r \times (2^{r-1} - 1)$

Proof. The removal of a cut edge e incident at the root vertex of T_r leaves T_r into 2 components with $2^{r-1} - 1$ and 2^{r-1} vertices. By Corollary 3, $\Pi(T_r) \geq 2^r \times (2^{r-1} - 1)$Let $e \in E(T_r)$. The congestion on e is $2|V(G_e)|\,|V(G_e')|$ where G_e and G_e' are the components of $T - e$. We now define a function $f: E(G) \to N$ by $f(e) = 2|V(G_e)|\,|V(G_e')| = 2|V(G_e)|\,(2^{r-1} - |V(G_e)|$. The function f is maximum when $\left|\frac{2^r - 1}{2}\right|$.

The following theorem is easy consequence of Theorem 4.

Theorem 5 : *Let* G *be a* k*-rooted complete binary tree* T_r^k. *Then* $II(G) = 2^{2r+1}\lceil k/2 \rceil \lfloor k/2 \rfloor$.

The sibling tree ST_r is obtained from the complete binary tree T_r by adding edges (sibling edges) between left and right children of the same parent node.

The 1-rooted sibling tree ST_r^1 is obtained from the 1-rooted complete binary tree T_r^1 by adding edges (sibling edges) between left and right children of the same parent node. The k-rooted sibling tree ST_r^k is obtained by taking k copies of vertex disjoint 1-rooted sibling tree ST_r^1 on 2^r vertices with roots say $r_1, r_2, \ldots, r_k$ and adding the edges (r_1, r_{i+1}) , $2 \leq i \leq k-1$.

Theorem 6 : *Let G be a sibling tree ST_r, Then $\Pi(G) = 2^{r-1} \times (2^{r-1} - 1)$.*

Proof. The removal of edges $S = \{e_1, e_2\}$ incident at the root vertex of ST_r leaves ST_r into 2 components with $2^{r-1} - 1$ and 2^{r-1} vertices. $R(a, b)$ By Lemma 2, $\Pi(ST_r) \geq 2 \times 2^{r-1} - 1 \times 2^{r-1}$. Let $R(a, b)$ denote a shortest path between a and b. If a is in G_i then it passes through edge e_i. Therefore congestion each of the on edge incident with root vertex is $(2^{r-1} - 1) \times 2^{r-1}$. The congestion on each edge is minimum. Among all the edges, cut edge e incident at the root vertex of ST_r is having maximum congestion.

The following theorem is easy consequence of Theorem 4

Theorem 7 : *Let G be a k-rooted sibling tree ST_r^k. Theorem $\Pi(G) = 2^{2r+1}\lceil k/2 \rceil \lfloor k/2 \rfloor$.*

5. Conclusion

We have obtained the edge-forwarding indices of cylinder, torus and certain trees. We also provide an linear time algorithm to solve it. The technique used in this paper is simple and elegant. It is also an interesting line of research to solve the edge-forwarding index problem for all classes of graphs like butterfly, circulant, chord and so on.

References

1. Xu, Topological Structure and Analysis of Interconnection Networks, Kluwer Academic Publishers, 2001.
2. F.K. Chung, E.G. Coffman, M.I. Reiman, B. Simon, The forwarding index of communication networks, IEEE Trans. Inform. Theory, 33 (1987), 224 – 232.
3. G. Gauyacq, Edge-forwarding index of star graphs and other Cayley graphs, Discrete Appl. Math. 80 (1997), 149 – 160.
4. M.C. Heydemann, J.C. Meyer, D. Sotteau, On the forwarding indices of networks, Discrete Appl. Math., 23 (1989), 103 – 123.
5. M.C. Heydemann, J.C. Meyer, J. Opatrny, D. Sotteau, Forwarding indices of k-connected graphs, Discrete Appl. Math., 37/38 (1992), 287 – 296.
6. M.C. Heydemann, J.C. Meyer, D. Sotteau, J. Opatrny, Forwarding indices of consistent routings and their complexity, Networks, 24 (1994), 75 – 82.
7. X. Hou, M. Xu, J. Xu, Forwarding indices of folded n-cubes, Discrete Appl. Math. 145 (2005), 490 - 492.
8. F. Shahrokhi, L.A. Szekely, Constructing integral uniform flows in symmetric networks with application to the edge-forwarding index problem, Discrete Appl. Math., 108 (2001), 175 – 191.
9. P. Sole, The edge-forwarding index of orbital regular graphs, Discrete Maths., 130 (1994), 171 – 176.
10. A. Bouabdallah, D. Sotteau, On the edge forwarding index problem for small graphs, Networks, 23 (1993), 249 – 255.
11. J.M. Xu, M. Xu, The Forwarding Indices of Graphs - a Survey, CoRR, 1204-2604, 2012.
12. I. Rajasingh, P. Manuel, B. Rajan, M. Arockiaraj, Wirelength of Hypercubes into trees, Discrete Appl. Math. 160 (2012), 2778 - 2786.