OPTIMUM INTERVAL ROUTING IN k-CATERPILLARS AND MAXIMAL OUTER PLANAR NETWORKS

Gur Saran Adhar
Computer Science Department
University of North Carolina at Wilmington, NC, USA 28403
email: adharg@uncw.edu

ABSTRACT
In this paper we present optimum interval routing algorithms for message passing networks that have topological structure defined by k-caterpillars and maximal outer planar graphs (MOPS). The routing algorithms are optimum in the sense that the route built between any source-destination pair is of minimum length and that the routing function executing at each node has time complexity which is linear in the size of the network.

KEY WORDS
Routing, Network, Algorithms, Message Passing

1 Introduction

Routing problem in message passing networks is solved by either maintaining at every node detailed routing information for all other nodes in the network (explicit routing) or by exploiting the information implicit in the labelling of the nodes and links (implicit routing). Interval routing, which is a type of implicit routing, was first introduce by Santoro and Khatib [1] and further studied by Van Leeuwen and Tan [2], Fraigniaud and Gavoille [3]. The most distinguishing feature of an interval routing scheme is that first a total order is placed on the node labels in the network and this order is used subsequently to route the messages.
Van Leeuwen and Tan [2] posed the following fundamental question with respect to interval routing in networks with arbitrary topology, - "Is there an optimal Interval Routing Algorithm for networks with arbitrary topology?" This question was subsequently answered in negative by Ruzicka [4]. Although, the question regarding optimal routing has been settled for networks with arbitrary topology, effort including this research, has been made to identify topologies which entertain optimal interval routing. Interval routing schemes for trees, complete graphs, rings, grids, and complete bipartite graphs appear in [2].
We describe optimum routing schemes for two network topologies: which can be described by (i). k-caterpillars (Section 2.1) and (ii). maximal outer planer graphs (MOPS) (Section 2.2).

2 Labelling and Interval Routing Algorithms

Optimum interval routing algorithm for k-caterpillars is presented in Section 2.1 and for Maximal Outer planar Graphs in Section 2.2.

2.1 k-Caterpillar

Our algorithm for k-caterpillars is based on the fact that when *k-leaves* are removed from a k-caterpillar the resulting graph is a *k-path.* The vertices in the alternating k-clique and (k+1)-clique of the k-path can then be ordered in a canonical sequence $\mathcal{L}$ starting from any one end of the k-path. This canonical sequence $\mathcal{L}$ of nodes in k-path forms the basis of the optimum interval routing algorithm.

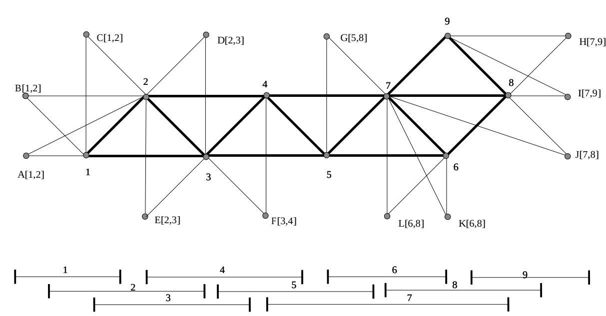

Figure 1. An example 2-caterpillar, labelled 2-path (highlighted) and interval representation of 2-path

Definition 2.1.1 *A* caterpillar *(also called Hamiltonian tree) is a tree which results into a path when all the leaves are removed.*

Definition 2.1.2 *A* k-caterpillar *is a k-tree (Definition 2.1.3) which results into a k-path (an alternating* *sequence of k and (k+1) complete subgraphs) when k-leaves (vertices with degree k) are removed.*

An example 2-caterpillar along with a 2-path (highlighted) are illustrated in the Figure 1.

Definition 2.1.3 *A k-tree is defined recursively as follows:*

- Basis: *A complete graph on k vertices (K_k) is a k-tree.*
- Recursive step: *joining a new vertex to a complete subgraph of order k (K_k) in the graph already constructed is a k-tree.*

Since k-trees and k-paths are triangulated graphs as well, they entertain a perfect elimination order [9], [10], [15], [16] on their vertices. In the case of k-path a canonical sequence $\mathcal{L}$, which is also an elimination order, can be defined on the vertices of the k-path. Furthermore, since a k-path is also an interval graph, it has a interval representation corresponding to the canonical sequence $\mathcal{L}$. The total order defined by $\mathcal{L}$ on the nodes in the k-path is consistent with the order of intervals in the interval representation when they are arranged by their right end points. For the example 2-caterpillar the interval representation corresponding to the 2-path is illustrated in Figure 1. In the illustration, the intervals are numbered by their right end points and the same labels also appear on the nodes in the 2-path. The interval representation of a k-path exhibits the following property.

Theorem 2.1.4 *In any (k+1)-clique C_m of a k-path, the interval j with rightmost right end point adjacent to an interval i in the interval representation, corresponds to the vertex j adjacent to vertex i that has the highest rank within the canonical sequence $\mathcal{L}$ among the vertices of C_m.*

Proof: (Sketch) The cliques of an interval graph can be arranged in a canonical sequence such that intervals appears in the consecutive cliques within the sequence.

Corollary 2.1.5 *In any (k+1)-clique C_m of a k-path, the interval j with leftmost left point adjacent to an interval i in the interval representation, corresponds to the vertex j adjacent to vertex i that has the lowest rank within the canonical sequence $\mathcal{L}$ among the vertices of C_m.*

This property of k-paths leads to a simple labelling and interval routing algorithm for k-caterpillars.

```
Labelling Algorithm_CAT
(Input: V,E; Output: L)
begin
  i  Delete all vertices with degree k (k-leaves);
  ii Compute canonical sequence L as follows:
       index ← 1;
       find a vertex v with degree(v) == k;
       L(v) ← index;
       while (V ≠ φ)
           index++;
           find a vertex w adjacent to v
           with degree(w) == k;
           L(w) ← index;
           v ← w;
       end while
end.
```

Once the labelling $\mathcal{L}$ is completed with the Labelling Algorithm the following Interval Routing Algorithm $Transmit_i$ performs transmission at node i of a message msg coming from some source vertex labelled n to any other destination vertex labelled m. $Transmit_i$ is a daemon process at node i which: (i). wakes up by the event signalling the arrival of a message msg; (ii). examines destination m in the header of the message; and (iii). transmits the message to the next node using the algorithm $Transmit_i$.

```
Interval Routing Algorithm_CAT
Transmit_i(Message : msg, To : m)
begin
  i   if (m == i) then
      DeliverAt(i); exit; /* message has reached
      destination */
      endif
  ii  if m ∈ adj(i) then
      DeliverAt(m); exit; /* message has reached a
      neighbor */
      endif
  iii if (m > i) then
        find the vertex j with highest rank in L
        adjacent to i
          j ← max{v | v ∈ adj(i)}
        DeliverAt(j);
      else /* (m < i) */
        find the vertex j with lowest rank in L
        adjacent to i
          j ← min{v | v ∈ adj(i)}
        DeliverAt(j);
      endif
end.
```

In the trivial case when the k-path is a 1-path (chain) each node transmits to either its left neighbor or to its right neighbor. In case of k-path, Theorem 2.1.4 provides the basis for the message to be delivered thru an intermediate node which has the rightmost (leftmost) right end (left end) in the interval representation. The time required to execute $Transmit_i$ at node i is constant and the route determined is shortest and therefore considered optimal.

2.2 Maximal Outer Planar Graphs

A planar graph G is *outer-planer* if and only if there is an embedding of G on the plane in which every vertex lies on the exterior face. A maximal outer planer graph (called MOP) is an outer planar graph with the maximum number of edges. That is, any additional edge will destroy its outer planar property.

Definition 2.2.1 *Given embedding of a planar graph on a plane, the* geometric dual *is a graph in which:*

- *the vertex set corresponds to faces of the planar graph in the embedding and*
- *there is an edge* (f_i, f_j) *between two vertices* f_i *and* f_j *of the geometric dual if the corresponding faces share a common edge.*

Removing the vertex corresponding to the exterior (unbounded) face of the planar graph results in the *weak dual* graph. In this paper however, we will refer to the weak dual graph simply as a dual graph. An example MOP and its dual graph are illustrated in Figure 2. We recall below some properties of dual graphs which are useful here.

Proposition 2.2.2 *[7] Dual graph of a MOP is a 3-regular tree (with vertex degree either 1 or 3).*

Proposition 2.2.2 combined with the separator theorem for trees [14] and the fact that all regions in a MOP are triangular, give us the following theorem.

Theorem 2.2.3 *There exists a triangular region* $\mathcal{R}$ *in the planar embedding, of a MOP with* n *vertices* ($n > 3$)*, the removal of* $\mathcal{R}$ *partitions the MOP into disjoint sets A, B, and C such that:* $\mid A \mid, \mid B \mid, \mid C \mid \leq \lceil \frac{n-3}{2} \rceil$ *and* $\mid A \mid + \mid B \mid + \mid C \mid = (n-3)$

Proof: Maximal outer planer graphs (MOPs) are Hamiltonian, in fact, the sequence of vertices and edges lying in the outer region of a planar embedding of a MOP is a Hamiltonian cycle. Therefore any edge (u, v) which does not lie on the the outer region is a separator. Also, since dual graph of the MOP is a tree there exists a triangular region R that correspond to a $(\frac{1}{2} - \frac{1}{2})$ tree separator node in the dual graph. Such a region R must have at least one edge (u, v) which does not lie on the outer region

Definition 2.2.4 Median *of a graph is a vertex that minimizes the* average *distance from any other vertex.*

It is shown in [13] that a vertex of a tree is a median iff neither of the subtrees attached to it contains more than one half of the vertices of the tree. Using this fact, following corollary of the Theorem 2.2.3 can be derived.

Corollary 2.2.5 *The median vertex* r *of the dual graph of the MOP is one of the three vertices bounding the separator region* $\mathcal{R}$ *defined by Theorem2.2.3*

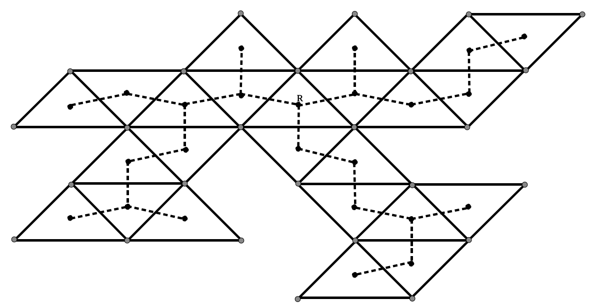

Figure 2. An example MOP, its dual graph (edges in dotted lines) and region $\mathcal{R}$

The optimality of our routing algorithm results from the fact that, by definition, the average distance from median to any other vertex is minimum. Therefore, if the root of the minimum distance spanning tree is made coincident with the median of the network, the routes will be such that the average distance over all the source-destination pairs will be minimized. Furthermore, Corollary 2.2.5 provides us with mechanism to identify the median in a MOP.

The main algorithm proceeds as follows. In the first step we identify the triangular region $\mathcal{R}$ (as defined by Theorem 2.2.3) in the planar embedding of a MOP which splits the graph almost evenly. Details of the first step appear in subsection 2.2.1. Since the average distance from the median to any other vertex is minimized we select median as the root for the minimum distance spanning tree. In the second step, we find the minimum distance spanning tree (mst) rooted at the median, which is one of the three vertices bounding the triangular region $\mathcal{R}$ (Corollary 2.2.5). The second step is described in detail within subsection 2.2.2. Finally the post-order indexing of the mst is performed in step three to obtain the labelling $\mathcal{L}$.

Labelling Algorithm_MOP
(**Input**: V,E; **Output**: $\mathcal{L}$)
begin

i Find the median vertex of the MOP
$C \leftarrow Median(V, E)$;
ii Find MST rooted at median C
$T \leftarrow Rooted\_MST(C, V, E)$;
iii Find post-order labelling of the rooted tree T
$\mathcal{L} \leftarrow post\_order(T)$;

end.

Once the labelling $\mathcal{L}$ is completed by the Labelling Algorithm, the Interval Routing Algorithm $Transmit_i$ performs transmission at node i of a message msg coming from some source vertex labelled n to any other destination vertex labelled m. $Transmit_i$ is a daemon process at node i which: (i). wakes up by the event signalling the arrival of a message msg; (ii). examines destination m in the header of the message; and (iii). transmits the message to the next node using the algorithm $Transmit_i$.
Starting from the source node every intermediate node will find a link α_s such that the interval $[\alpha_s, \alpha_{s+1}]$ defined by labelling $\mathcal{L}$ contains the destination m. The intermediate node then sends the message down the link labelled α_s and the whole process is repeated. We use $S_T(i)$ in the algorithm to denote the set of successor nodes of x in tree T

Interval Routing Algorithm_MOP
$Transmit_i(Message : msg, To : m)$
begin

i **if** $(\mathcal{L}(m) == \mathcal{L}(i))$ **then**
DeliverAt(i); exit; /* the message has reached its destination.*/
endif
ii **if**$(\mathcal{L}(m) > \mathcal{L}(i))$ **then**
Set i to the label of predecessor;
$Transmit_i(msg, m)$
else /*$(\mathcal{L}(m) < \mathcal{L}(i))$*/
find label α_s in $\mathcal{L}$ among the nodes $S_T(i)$
such that $\alpha_s \leq m \leq \alpha s + 1$;
Set i to the label of successor
reached over link α_s;
$Transmit_i(msg, m)$

end.

The time required to execute $Transmit_i$ at node i is dominated by step-(ii) and is linear in the size of the network. The average length of routes between source destination pairs is minimum and therefore the algorithm is considered optimal.

2.2.1 Median of the dual graph of a MOP

Median C of the MOP graph (V, E) is found in step-(i) of the Labelling Algorithm_MOP by navigating over the dual graph, which is a tree, of the MOP. We assume that MOP is specified by pairs (e, f) whenever an edge e belongs to a face f. In the following algorithm we use $S_T(f)$ to denote the set of successor nodes of f in tree T and $P_T(f)$ to denote the predecessor node of f in tree T.

Algorithm Median_MOP
(**Input**: V,E; **Output**: C)
begin

i Construct the plane dual tree $T^\dagger$ as follows:
(i). sort according to (e, f) where e is an edge of the face (region) f.
(ii). create edge (f_1, f_2) for every pair (e, f_1) and (e, f_2).
(iii). sort edges (f_1, f_2) and delete duplicates.
ii Make $T^\dagger$ a directed tree by selecting an arbitrary node for the root.
iii For each node f of $T^\dagger$ compute:

$$size(f) = 1 + \sum_{x \in S_{T^\dagger}(f)} size(x)$$

iv Identify a node r of $T^\dagger$ such that:
$size(r) \leq \frac{n}{2}$ and
$size(P_{T^\dagger}(r)) \geq \frac{n}{2}$
return (r);

end.

2.2.2 Rooted Minimum Distance Spanning Tree of MOP

Our method for determining the minimum distance spanning tree (mst) for the MOP in step-(iii). of the Labelling Algorithm_MOP depends on a invariant property of *recursive labellings* in k-trees (Definition 2.2.2.1) of which MOPs are a proper subclass.

Definition 2.2.2.1 *A* recursive labelling *of a k-tree with n vertices uses integers* 1 *through* k *to label* base vertices *and label* $(k + 1)$ *through* n *to label remaining vertices in the order they are added in a recursive construction process.*

It should be noted here that a k-tree can have many different recursive labelling, however the property stated in Lemma 2.2.2.2 and in Lemma 2.2.2.3 are invariant among all the labelling of a k-tree. That is, any recursive labelling of a k-tree orders adjacent (nonbasic) vertices concordantly with nondecreasing value of the distance to the root. We recall here the following Lemmas from [5] to confirm this fact.

Lemma 2.2.2.2 *[5] Any two adjacent, nonbasic vertices u and v of a k-tree Q, with a given base structure must be labelled in the same manner in all recursive labelling of Q.*

Lemma 2.2.2.3 *[5] In any recursive labelling of a k-tree Q if* $label(u) < label(v)$*, then their shortest distance to the root r of Q satisfy the inequality* $d(u, r) \leq$

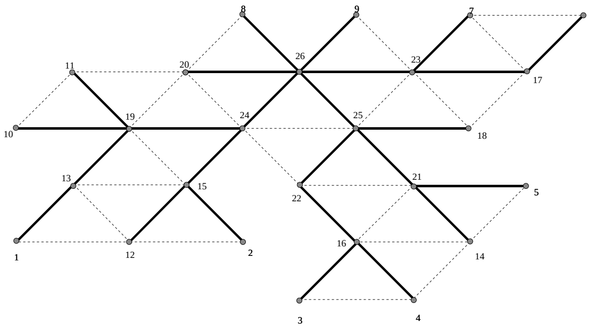

Figure 3. Example MOP after labelling, MST rooted at the median vertex (highlighted)

$d(v, r)$. *Furthermore, the addition of a simplical vertex does not change the shortest distance to the root of any other vertex.*

Since MOPs are planar 2-trees (k-trees with k=2) it is easy to verify that the Algorithm Rooted_MST_MOP below correctly finds the minimum distance spanning tree of a rooted MOP.

Algorithm Rooted_MST_MOP
(**Input**: C, V, E; **Output**: T)
begin

- i Compute any recursive labelling $\mathcal{L}$ for the k-tree Q rooted at r;
- ii For each vertex v find its neighbor u with smallest rank in the labelling $\mathcal{L}$;
- iii Add edge (v, u) to T;

end.

For the example maximal outer planer graph in Figure 2 the minimum distance spanning tree rooted at the median vertex (highlighted) is illustrated in Figure 3.

3 Conclusion

We have presented optimum interval routing schemes for two classes of networks which can be represented by graphs that exhibit recursive structure. The routing function requires constant time for the k-Caterpillars and has linear time complexity for the Maximal Outer Planar (MOP) Networks. Furthermore, the route determined by the algorithms are of smallest length. In this respect the interval routing is considered optimum.

References

[1] Nicola Santoro and Ramez Khatib, Labelling and Implicit Routing in Networks, The Computer Journal, 28(1), 5-8 1985.

[2] J. Van Leeuwen and R.B. Tan, Interval Routing, The Computer Journal, 30(4), 298-307 1987.

[3] P. Fraigniaud and C. Gavoille, Interval Routing Schemes, Algorithmica, 21, 155-182 1998.

[4] P. Ruzicka, A Note on the Efficiency of Interval Routing Algorithm, The Computer Journal, 34(5), 475-476 1991.

[5] Anderzej Proskurowski, Recursive Graphs Recursive Labellings and Shortest Paths, SIAM Journal of Computing, 10(2), 391-397 1981.

[6] Anderzej Proskurowski, Separating Subgraphs in k-trees: Cables and Caterpillars, Elsevier Science Publishers B.V.(North Holland), 275-283 1984.

[7] Sandra M. Hedetniemi, Anderzej Proskurowski and Maciej M. Syslo, Interior Graphs of Maximal Outerplane Graphs, Journal of Combinatorial Theory, Series B 38, 156-167 1985.

[8] Sandra M. Mitchell, Interior Graphs of MOPS, Tech. Rep., Dept. of Computer Science Univ. of Oregon, CS-TR-79-2.

[9] L.W. Beineke and R.E. Pippert, Properties and Characterization of k-trees, Mathematika, 18, 141-151 1971.

[10] Donald J. Rose, On Simple Characterization of k-trees, Discrete Mathematics, 7, 317-322 1974.

[11] O. Gerstel and S. Zaks, A New Characterization of Tree Medians with Applications to Distributed Algorithms, Tech. Rep., Dept. of Computer Science, Technion, Hiafa, 1991.

[12] Dexter C. Kozen, The Design and Analysis of Algorithms, Springer Verlag Pub., 1991.

[13] B. Zelinka, Medians and peripherians of trees, Arch. Math. (Brno), 87-95, 1968.

[14] F.R.K. Chung, Separator Theorem and Their Applications, Paths Flows and VLSI Layout, Springer Verlag Pub., 1991.

[15] Joseph Naor, Moni Naor and Aljandro Schaffer, Fast Parallel Algorithms for Chordal Graphs, *FOCS-1987*, 355-364.

[16] D.J. Rose, Triangulated Graphs and the elimination process, *J. Math. Anal. Appl.*, VOL. 32,(1970), 597-609.

[17] Charbel Farhat, On the mapping of Massively Parallel Processors onto Finite Element Graphs, *Computers and Structures*, VOL. 32,(1989), 347-353.