Drawing Telecommunication Networks

Ioannis G. Tollis and Chunliang Xia

Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75083-0688
tollis@utdallas.edu

Abstract. The design and analysis of cost effective *survivable telecommunication networks* is a very important problem. We study techniques for visualizing survivable telecommunication networks. The visualization of telecommunication networks is very useful in aiding the design process of minimum cost networks and the management of network operations. We present several linear time algorithms for drawing telecommunication networks (with optimal area) so that important properties are displayed. Given a ring cover of a network, our algorithms display it in such a way that rings are easily identifiable and possible problems can be spotted by network designers.

1 Introduction

The problem of drawing a graph in the plane has received increasing attention recently due to the large number of applications [1]. Examples include VLSI layout, algorithm animation, visual languages, and CASE tools [2]. Vertices are usually represented by points and edges by simple open curves. In this paper we study techniques for visualizing telecommunication networks. The visualization of telecommunication networks is very useful in aiding the design process of minimum cost networks and the management of network operations [7]. We present linear time algorithms for drawing survivable telecommunication networks (with optimal area) so that important properties are displayed.

The design and analysis of cost effective *survivable telecommunication networks* is a very important problem [3, 5, 6, 9, 10, 12]. Most problems that aim towards minimizing the total cost of such a network are NP-hard [4]. For that matter, computer tools to aid the design and analysis of telecommunication networks are needed. A central problem of such tools is how to draw a network on the computer screen such that important aspects of the network can be easily captured and an improved solution can be obtained by a user interactively.

The problem is defined as follows: Let $G = (V, E)$ be a telecommunication network with a set of nodes V (representing the sites of switches) and a set of links E (representing the electrical wires or optical fiber links between nodes). The traffic requirements between the nodes are defined by an $n \times n$ matrix T, where $T(i, j)$ corresponds to the amount of traffic between nodes i and j. We need to design a network which (a) satisfies the traffic requirements, (b) can survive failures, and (c) the cost of the network is minimum. A network is *1-survivable* if it can survive the failure of a link e, i.e., the removal of link e does not disconnect the network and the traffic that originally travels through e can be accomodated on another path. The multi-ring architecture is considered as a cost-effective survivable network architecture due to its simplicity, improved survivability and bandwidth sharing [11, 12]. A *ring cover* of G is a set of rings (cycles) of G such that the rings are connected and every node in V is included in at least one ring. Apparently, a network with a ring cover is 1-survivable since the switches automatically send the required traffic around the ring if a link failure occurs [10, 11, 12].

Since the nodes of the network correspond to sites, they have geographic coordinates. Hence, the network can be drawn naturally with little effort. However, the important properties of the network that designers are interested in (such as rings) are not displayed. In this paper, we present algorithms for drawing telecommunication networks in order to aid the design of cost-efficient networks. Given a ring cover of a network, our algorithms display it in such a way that rings are easily identifiable and possible problems can be easily spotted by network designers. An example of a drawing of a ring cover is shown in Fig. 1. Our algorithms run in linear time and produce drawings that require optimal area.

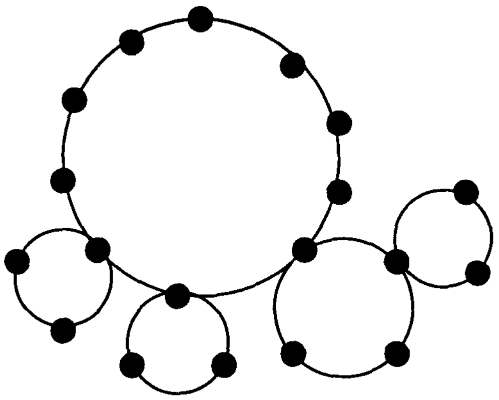

Fig. 1. Example of a drawing of a ring cover.

The rest of the paper is organized as follows: In Section 2, we discuss the criteria for drawing ring covers. Section 3 is devoted to the description of several algorithms that draw a telecommunication network, given a ring cover. In Section 4 we summarize the results and discuss some open problems.

2 Preliminaries

Ideally, we would like to represent every ring as a cycle, see Fig. 1. However, due to the complexity of interactions among rings in a ring cover, it is not always possible to draw rings as cycles. For example, if three rings share a common node then it is impossible to draw three disjoint cycles. If such a case occurs, we shall use a geometric shape with a slight deviation from a cycle, called *almost cycle*, to represent rings. An almost cycle is a geometric shape that, except for a few nodes, all of its nodes are placed on the perimeter of a cycle.

As is the case in most graph drawing algorithms, the existence of unnecessary crossings is viewed as harmful to the readability of the drawing. This is also true in our ring cover drawings, since unnecessary crossings may visually create rings that do not exist in the original ring cover. Thus, minimizing such crossings is central to our approach. We also assume the existence of a *resolution rule*, that is, in the final drawing, any two nodes of the network must be kept far enough so that the human eye can tell them apart. This implies that the drawing cannot be arbitrarily scaled down. If we honor such a rule in our drawings and represent each ring as a cycle, there is a trivial lower bound of $\Omega(N^2)$ on the area required for the drawing, where N is the number of nodes in the network. This happens when there is only one ring in the ring cover.

Drawing ring covers in general is rather difficult. Even if we relax the constraint of using cycles to represent rings, and allow ourselves to use any convex polygon to represent a ring, not all ring covers admit such representation. In order to capture the complexity of interactions among rings, a new graph G' is introduced. Given a network G and its ring cover C, a *contact node* is a node of G that is contained in at least two rings. Let V' be the collection of contact nodes, $G' = (V' \cup C, E')$, where $E' = \{(R, c) \mid R \in C, c \in V' \text{ and } c \text{ is in } R\}$. Graph G' is called *ring-contact node graph.* The definition of a ring cover implies that G' is a connected graph.

Denote a ring as an ordered sequence of its nodes $R = \{n_1, n_2, ..., n_p\}$ in clockwise order. Let $R_1 = \{n_1, n_2, ..., n_p\}$ and $R2 = \{m_1, m_2, ..., m_q\}$ be two rings, and suppose that they have k contact nodes, say $\{c_1, c_2, ...c_k\}$, where c_i appears before c_{i+1} in the clockwise order around R_1. We say that R_1 and R_2 are *compatible* if and only if there is a cyclic shift of the nodes of R_2, so that contact nodes $\{c_1, c_2, ...c_k\}$ appear in this order, or in the reverse order (i.e.,$\{c_k, ..., c_2, c_1\}$). The following lemma describes a necessary condition for a ring cover that admits a convex polygon representation.

Lemma 1. *Given a ring cover $C = \{R_1, R_2, ...\}$, if C has a convex polygon representation then all rings in C are pairwise compatible.*

As pointed out earlier, ring cover drawing is very difficult or even impossible when G' is an arbitrary graph. So, we focus our attention to some special cases. In the rest of the paper, we assume that G' is a tree, which is often the case in practice. If G' is a tree then any two adjacent rings share exactly one contact node.

3 Drawing Tree-Structured Ring Covers

In this section we present algorithms for drawing tree-structured ring covers of networks. The ring-contact node graph is a tree, which will be called T. We propose three standards for drawing ring covers:

1. Outside Drawings: Each ring is drawn outside of the other rings.
2. Inside Drawings: Each ring (except for the root) is drawn inside some other ring(s).
3. Mixed Drawings: A ring may be drawn outside or inside other rings.

We will show that there are advantages and disadvantages to each technique.

3.1 Outside Drawings

It is natural to draw two rings next to (and outside of) each other when they share a contact node. Since the structure of the ring-contact point tree T may vary, and the number of nodes contained in different rings could be drastically diverse, it is important to position each cycle in a proper place. Otherwise, there will be some crossings.

For a ring node u and all its subtree in the ring-contact node tree T, define a cycle R_u, called the *enclosing cycle* of u, such that R_u is large enough that we can place the drawing of u and its subtree inside R_u. Our intention is to draw u and its subtree in a confined area, so that it does not intersect rings of other subtrees of T.

Clearly, any ring node u (except for the root) of T has a parent. In the drawing of u, we place u's contact node with its parent on the top half of the cycle, and place all other nodes (including all other contact nodes) at the bottom half of the cycle. By doing so, we avoid intersections between the ancestors of u and the children of u, see Fig. 2. The cycle drawn with dotted line is the enclosing cycle. Assume that ring node u, has i children, $u_1, u_2, ..., u_i$ in T. If all the rings R_j of u_j, for $j = 1, 2, ..., i$, have been drawn, we can draw cycle R, corresponding to u, and place R and R_j in the following fashion:

First we draw a cycle R' and we place all the nodes of u and all the cycles of R_j, $j = 1, 2, ..., i$, evenly so that the center of each R_j lies on the perimeter of R'. The distance among the nodes and the cycles is the distance required by the resolution rule. It is easy to see that the diameter of R' should be larger than the diameter of any R_j.

Next, we draw a new cycle R with radius twice the radius of R', and place all R_j, $j = 1, 2, ..., i$, as well as all the nodes of u, except u's contact node with its parent, on the bottom half perimeter of R. This can be achieved easily by cutting the perimeter of R' at the point of u's contact node with its parent, and using that as the bottom half of perimeter of R. This is possible since the radius of R is equal to the diameter of R'. Finally, we pull each R_j downward until it touches R at their common contact node. Note that the resolution rule is preserved during this process. It is shown in [8] that a convex polygon with n vertices needs

$\Omega(n^3)$ area to be drawn with integer coordinates. The coordinates of the nodes are naturally computed using a polar coordinate system, and they are stored as floating point numbers. This is the case for the other drawing techniques as well. Using the procedure described above, we obtain the following lemma.

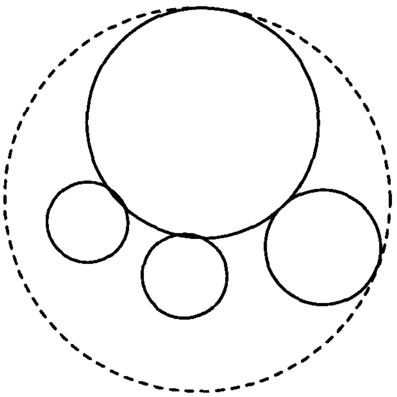

Fig. 2. Drawing of a ring node and its enclosing cycle.

Lemma 2. *For any given ring node u and all its subtree in T, the enclosing cycle R_u of u has radius $O(N)$, where N is the number of nodes in u and its subtree.*

In the above description, we implicitly assume that no more than two rings share a contact node. This is not true in general. If three or more rings share a contact node, we could pretend that they have distinguished contact nodes, but in our final drawing, instead of drawing them as cycles, we can draw them as almost cycles. The handling of such a case is illustrated in Fig. 3.

The algorithm that produces an outside drawing of a ring cover essentially traverses T in postorder, while drawing each ring as it visits the corresponding node.

```
Algorithm OutDraw (v)
Input: Ring cover $C$ and ring-contact node graph $G'$ with root $v$.
Output: Outside drawing of $C$.
begin
  1 if $v$ is a leaf, then draw the ring corresponding to $v$; return;
  2 if $v$ is a node representing a ring, then
      • for all its children $u_i$ do OutDraw ($u_i$);
      • draw $v$ and its corresponding enclosing cycle as described in Lemma 2;
```

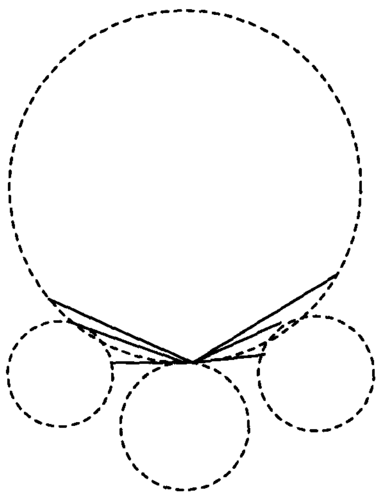

Fig. 3. Drawing almost cycles.

```
3 if $v$ is a node representing a contact node then
    • for all its children $u_j$ do OutDraw ($u_j$);
end
```

There are two modifications to the above algorithm that could result in a better utilization of the area. If v is a ring node and v has only one child, we could draw it as shown in Fig. 4(a); if v is the root of T, we could relax the constraint that all nodes be placed at the bottom half of the cycle, and we can place them evenly around the cycle of the root, as shown in Fig. 4(b).

Theorem 3. *Given a ring cover C and its corresponding ring-contact node tree T, Algorithm OutDraw produces an outside drawing for ring cover C, such that:*

1. *all the rings are drawn as cycles or almost-cycles;*
2. *there are no crossings;*
3. *the area of the produced drawing is at most $O(N^2)$, where N is the number of nodes in ring cover C;*
4. *the time complexity of Algorithm OutDraw is $O(N)$.*

3.2 Inside Drawing

In this section we consider *inside drawing* as an alternative to outside drawing. Instead of placing two rings side by side when they share a contact node, we place one ring inside the other. In other words, we always place rings of children

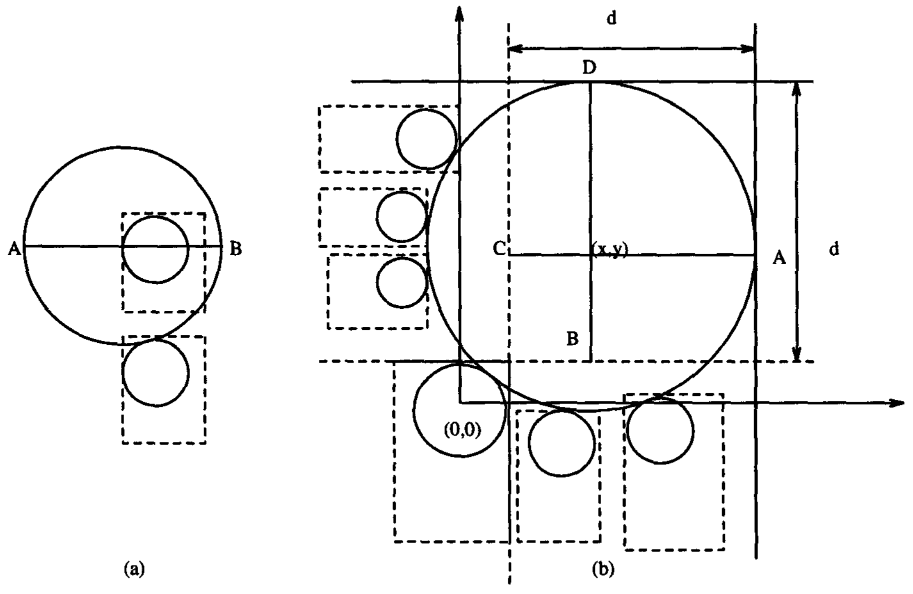

Fig. 4. (a) v has only one child; (b) v is the root of T.

inside the ring of their parent. Thus, the enclosing cycle for a ring node u is actually the ring for u itself, since all of its children are placed inside u.

Let u be a ring node in T and assume that u has i children, $u_1, u_2, ..., u_i$. If all the rings R_j of u_j, for $j = 1, 2, ..., i$, have been drawn, we can draw a cycle R, corresponding to node u, and place all the R_j's in R in the following fashion.

First, we draw a cycle R', and place all the nodes of u and all the cycles R_j evenly on the perimeter of R'. The distance between the nodes and the cycles is equal to the minimum distance required by the resolution rule. Each R_j is placed so that its center is placed on the perimeter of R, see Fig. 5. Clearly, the diameter of R' is chosen to be larger than the maximum diameter of the R_j's.

Next, we draw R using as center the center of R' and with radius large enough to enclose R' and the R_j's. Notice that the radius of R may be as large as two times the radius of R'.

Finally, we move each R_j towards the perimeter of R, such that, each R_j has one point of contact with R. This is their common contact point. Note that the resolution rule is preserved during this process.

By the above procedure we have the following lemma:

Lemma 4. *For any ring node u and its subtree in T, the radius of the corresponding cycle R is $O(N)$, where N is the number of nodes in u and its subtree.*

In outside drawing, if more than three rings share a contact node, we have to use an almost cycle to represent them. In inside drawing, we draw all of

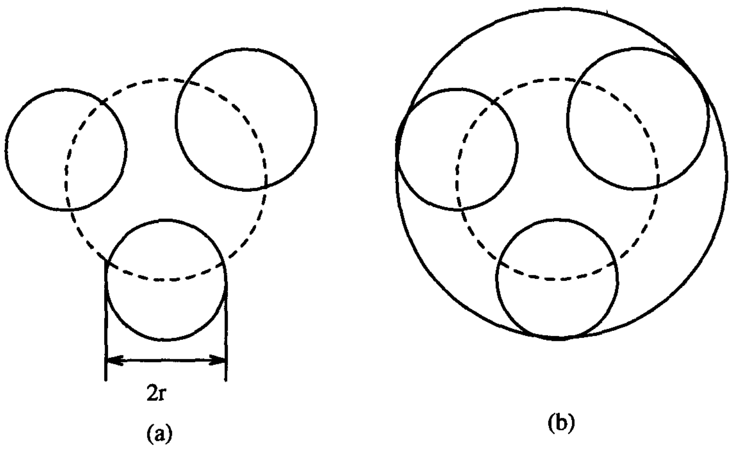

Fig. 5. (a) Placing the centers of rings R_J on the perimeter of R'; (b) Drawing R that encloses the R' and the R_j's.

them as cycles, and place one inside another. The following algorithm is used to draw a ring cover in inside fashion. Essentially, the algorithm travels the tree in postorder; after drawing all the children of a node recursively, the algorithm draws the parent node.

```
Algorithm InsideDraw ($v$)
Input: Ring cover $C$, and its ring-contact node tree $T$ with root $v$.
Output: An inside Drawing of $C$.
begin
  1 if $v$ is a leaf then draw a cycle corresponding to $v$; return;
  2 if $v$ is a node representing a ring then
      • for all its children $u_i$ do InsideDraw ($u_i$);
      • Draw $v$ as in the previous discussion or as shown in Fig. 6(a);
  3 if $v$ is a node representing a contact node and $v$ has only one child $u'$ then
    InsideDraw ($u'$);
  4 if $v$ is a node representing a contact node and $v$ has more than one children
    then
      • perform a transformation as shown in Fig. 6 (b);
      • InsideDraw ($T_1$);
end
```

If there are more than three rings that share a contact node, the transformation performed in Step 4 of the algorithm will draw a cycle that corresponds to a subtree inside the cycle that corresponds to another subtree so that they share the common contact point.

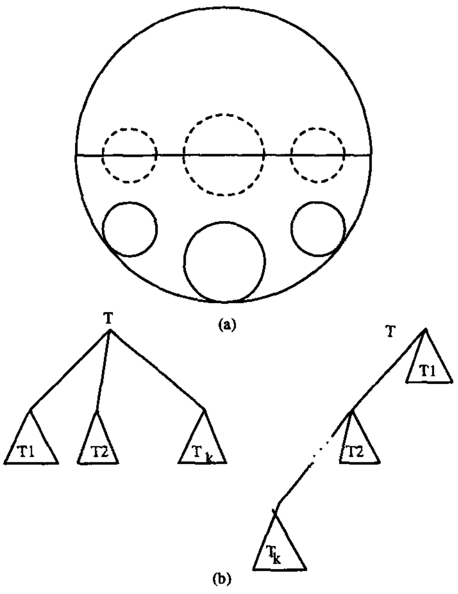

Fig. 6. Illustration of algorithm InsideDraw.

We have following theorem.

Theorem 5. *Given a ring cover C and its corresponding ring-contact node tree T Algorithm InsideDraw produces a ring cover drawing such that:*

1. *each ring is represented by a cycle;*
2. *there are no crossings;*
3. *the area required by the drawing is $O(N^2)$, where N is the number of nodes in C;*
4. *the time complexity of the algorithm is $O(N)$.*

It is interesting to observe that the order of the area occupied by the drawing does not depend on the starting point. In other words, the order of the area remains unchanged no matter which node is considered as the root of the tree.

So far, we have introduced two kinds of drawings for ring covers. On certain occasions, one kind of drawing is better than the other in terms of area. Fig. 7 shows an example where an inside drawing outperforms an outside drawing. Since the original ring R_0 is removed from the inside drawing, the actual drawing can be shrunk. Thus the area needed for the inside drawing is far less than that needed for the outside drawing. Fig. 8 shows an example containig five rings, where using an outside drawing is far better than an inside drawing. Clearly, an inside drawing of this example will take as much as $O(N^2)$ area, while an

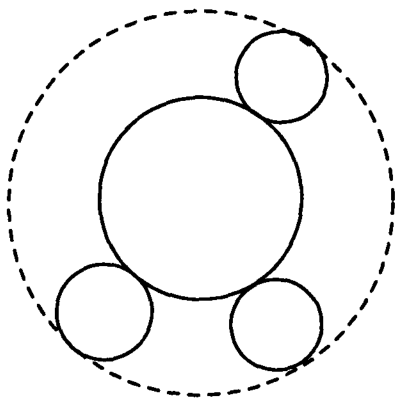

Fig. 7. An example where inside drawing is better than outside drawing.

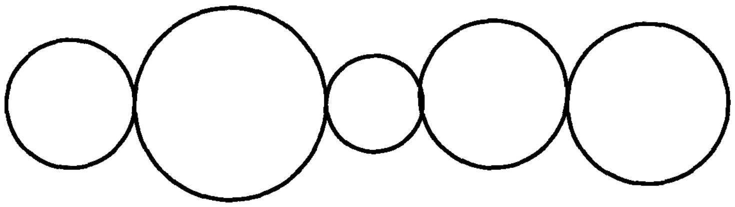

Fig. 8. An example where outside drawing is better than inside drawing.

outside drawing will take $O(N \times n_{max})$, where n_{max} is the number of nodes in the maximum ring. These facts motivate us to blend the two drawing techniques in order to obtain a mixed drawing.

3.3 Mixed Drawing

In this section we present a simple technique that attempts to break a given ring cover into smaller parts, such that each of the smaller parts will be drawn using the inside drawing technique. Observing Fig. 8, we realize that if the centers of all the rings are aligned on a straight line, the outside drawing will require less area than an inside drawing. So, we attempt to break a given ring cover into k subcovers, and draw each subcover using our inside drawing algorithm. Next we align all the centers of the outermost cycles of each subcover on a straight line. By our inside drawing algorithm, we know that the radius of each outermost cycle is proportional to the number of nodes contained in its corresponding subcover. Thus the final area required is $O(k \times {n_{max}}^2)$, where n_{max} is the number of nodes contained by the subcover with maximum number of nodes.

In order to obtain a better area utilization, we try to minimize n_{max}. In the following, we propose a heuristic to break a given cover into several subcovers, and then draw them separately using Algorithms InsideDraw and OutDraw.

```
Algorithm MixDraw()
Input: A ring cover $C$ and its ring-contact node tree $T$.
Output: Mixed drawing of $C$.
begin
  1 Assign a weight to each node in $T$, if node $u$ is a ring node, the weight of $u$
    is equal to the number of nodes in the ring corresponding to $u$; otherwise,
    the weight is zero;
  2 Find a longest path $l$ in $T$;
  3 Path $l$ obtained in Step 2 has the form $l = \{r_1, c_1, r_2, c_2, ...., r_k\}$, where $r_i$ is a
    ring node and $c_i$ is a contact node. Construct a partition $P = \{C_1, C_2, ..., C_k\}$
    of $C$ as follows:
      1. $C_i$ contains $r_i$ and its subtrees excluding the ones which are on $l$, for
         $i = 1, 2, ..., k$;
      2. suppose $c_i$ has children $q_1, q_2, ..., q_m \neq r_{i+1}$; w.l.o.g, assume $w(q_1) \geq$
         $w(q_2) \geq, ..., w(q_m)$.
      3. for $j = 1$ to $m$ do
         if $(w(r_i) > w(r_{i+1}))$ then $C_{i+1} = C_{i+1} \cup q_j, w(C_{i+1}) = w(C_{i+1}) + w(q_j)$;
         else $C_i = C_i \cup q_j, w(C_i) = w(C_i) + w(q_j)$.
  4 Use Algortihm InsideDraw to draw subcovers $C_1, C_2, ..., C_k$ respectively.
  5 Place outermost cycles of $C_1, C_2, ..., C_k$ on a straight line so that their centers
    are horizontally aligned and their contact points coincide.
end.
```

It is easy to see that every ring is represented by a cycle in a mixed drawing. There are no crossings, since we use only inside and outside drawings. In the worst case, the area required is $O(N^2)$, but the algorithm should be able to do better than that on the average. The time complexity is dominated by the ring cover partition, which takes linear time. Hence the time complexity is $O(N)$. By the above discussion we have following theorem.

Theorem 6. *Given a ring cover C and its corresponding ring-contact node tree T, Algorithm MixDraw produces a mixed drawing for ring cover C, such that:*

1. *the required area is $O(N^2)$, where N is the number of nodes in ring cover C;*
2. *there are no crossings;*
3. *all rings are drawn as cycles;*
4. *the time complexity of the algorithm is $O(N)$.*

4 Conclusions and Open Problems

We have presented three techniques for drawing ring covers of survivable telecommunications networks. Our algorithms run in linear time and they produce drawings that require $O(N^2)$ area. There are many open problems remaining:

1. Are there other reasonable ways to draw ring covers?
2. How can we draw ring covers when two rings share more than one contact node?
3. Are there other conditions that describe when a given ring cover does not admit such drawings?
4. How can we draw ring covers if the ring-contact node graph is not a tree?

References

1. G. Di Battista,P. Eades, R. Tamassia and I. G. Tollis, "Algorithms for Automatic Graph Drawing: An Annotated Bibliography," Dept. of Comp. Science, Brown Univ., Technical Report, 1993. To appear in *Comput. Geom. Theory Appl.* Preliminary version available via anonymous ftp from wilma.cs.brown.edu(128.148.33.66), files /pub/gdbiblio.tex.Z and /pub/gdbiblio.ps.Z.
2. G. Di Battista, E. Pietrosanti, R. Tamassia, and I.G. Tollis, "Automatic Layout of PERT Diagrams with XPERT," *Proc. IEEE Workshop on Visual Languages (VL'89)*, pp. 171-176, 1989.
3. G.R. Dattatreya, J.P. Fonseka, K. Kiasaleh, I.H. Sudborough, I.G. Tollis, and S. Venkatesan, "Advanced Network Topologies for Network Survivability," Technical Report, UTD, January 1993.
4. M.R. Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, 1979.
5. L.M. Gardner, M. Heydari, J. Shah, I.H. Sudborough, I.G. Tollis, and C. Xia, "Techniques for Finding Ring Covers in Survivable Networks," *Proceedings of IEEE GLOBECOM 1994*, to appear.
6. W.D. Grover, B.D. Venables, J.H. Sandham, and A.F. Milne, "Performance Studies of a Slef-Healing Network Protocol in Telecom Canada Long Haul Networks," *Proceedings of IEEE GLOBECOM 1990*, pp. 403.3.1-403.3.7.
7. G. Kar, B. Madden and R. S. Gilbert, "Heuristic layout Algorithms for Network Management Presentation Services" *IEEE Network*, November, 1988.
8. Y. Lin and S. Skiena, "Complexity aspects of visibility graphs" Technical Report 92/08 SUNY Stony Brook, 1992.
9. H. Sakauchi, Y. Nishimura, and S. Hasegawa, "A Self-Healing Network with an economical Spare-Channel Assignment," *IEEE GLOBECOM 1990*, pp. 403.1.1-403.1.6.
10. J. C. Shah, "Restoration Network Planning Tool", Proc. 8th Annual Fiber Optic Engineers Conf. April 21, 1992.
11. T.H. Wu, D.J. Collar, and R.H. Cardwell, "Survivable Network Architectures for Broadband Fiber Optic Networks: Model and Performance Comparisons," *IEEE Journal of Lightwave Technology*, Vol. 6, No. 11, November 1988, pp. 1698-1709.
12. Tsong-Ho Wu and R. C. Lau, "A Class of Self-Healing Ring Architectures for SONET Network Applications, *IEEE Trans. on Communication*, Vol. 40, No. 11, Nov. 1992.