Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT•

Computer Communications 26 (2003) 718–722

computer communications

www.elsevier.com/locate/comcom

Optimal placement of wavelength converters in trees, tree-connected rings, and tree of rings

Peng-Jun Wan*, Ophir Frieder, Liwu Liu

Department of Computer Science, Illinois Institute of Technology, USA

Received 7 August 2002; accepted 7 August 2002

Abstract

In wavelength routed optical networks, wavelength converters can potentially reduce the requirement on the number of wavelengths. A recent study [Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms (1998)] raised the following problem: choose a minimum number of nodes in a WDM network to place wavelength converters so that any set of paths requires the same number of wavelengths as if wavelength converters were placed at all node. This problem is referred to as minimum sufficient set problem. It was shown to be NP-complete in general WDM networks [Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms (1998)], and be as hard as the well-known minimum vertex cover problem [Proceedings of 10th ACM-SIAM Symposium on Discrete Algorithms (1999)]. In this paper, we extend their study in trees, tree-connected rings, and tree of rings which are widely used topologies in the telecommunications industry. We show that the optimal wavelength converter placement problem in these two practical topologies are tractable. Efficient polynomial-time algorithms are presented.

Keywords: Wavelength routed optical network; Wavelength converter; Minimum vertex cover; Tree; Tree-connected rings; Tree of rings

1. Introduction

In wavelength routed WDM (wavelength-division multiplexed) optical networks [11] without any wavelength conversion [12], the wavelength assignment must meet the wavelength continuity constraint, i.e. the same wavelength is allocated on all of the links in the path established for a connection [1,2,9,10,13,15]. Such constraint can be relaxed when wavelength converters are placed at certain nodes. If a node of the network contains a wavelength converter, any path that passes through this node may change its wavelength. In a network with wavelength converters, the wavelengths are assigned to individual links of all paths, with the restriction that the same wavelength is allocated on all of the links in any subpath that does not pass through a wavelength converter. Clearly wavelength assignments in networks with wavelength converters can be more efficient (i.e. use fewer wavelengths) than wavelength assignments for the same set of paths when no wavelength converters are available. One extreme example is that if each node has a wavelength converter, the number of wavelengths required for any routing is reduced down to the natural *congestion* or *load* bound, defined to be the maximum number of paths passing through any one link in the network.

However, it is not always necessary to place a wavelength converter at each node such that the number of wavelengths required by any set of paths is equal to its link load. As observed in Ref. [16], it is sufficient to place a converter at a single arbitrary node in a WDM ring to achieve this objective [16]. Motivated by this observation, the following question was raised in Ref. [16]: in a general WDM network, how to choose a minimum number of nodes to place wavelength converters so that the number of wavelengths required by any set of paths is equal to its link load? A set S of nodes in a network is said to be *sufficient* [16] if placing converters at the nodes in S can ensure that the number of wavelengths required by any set of paths equals to its link load. The *minimum sufficient set problem* is NP-complete in general WDM networks [16]. Furthermore, it is as hard as the *minimum vertex cover (MVC) problem* in undirected graphs [8]. Since providing an approximation ratio better than 2 for the MVC problem is a long-standing

* Corresponding author.
E-mail addresses: wan@cs.iit.edu (P.-J. Wan), ophir@cs.iit.edu (O. Frieder), liwuliu@cs.iit.edu (L. Liu).

0140-3664/03/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0140-3664(02)00206-2

open problem, this indicates that improving on the performance guarantee of two for minimum sufficient set problem will be difficult as well.

While the work in Ref. [8] provided approximation solutions for general WDM networks, we notice that the topologies of most practical WDM networks are not general. In particular, trees, tree-connected rings and tree of rings are of more practical concrete relevance to the telecommunications industry. For practical reasons, backbone telecommunication networks need to reflect irregularity of geography, non-uniform clustering of users and traffic, hierarchy of services, dynamic growth, etc. In addition, wide-area multiwavelength technology is evolving around current signal wavelength networking architectures and existing fiber networks. These are mainly SONET rings and tree-like interconnection of such rings [3,4,14]. In this paper, we will show that the minimum sufficient set problem in these special topologies can be solved in polynomial time. Our algorithms are based on the reduction of the minimum sufficient set problem to the MVC problem established in Ref. [8]. They are very efficient and easy to implement.

The remaining of this paper is arranged as follows. In Section 2, we first introduce some basic terminologies and the reduction from the minimum sufficient set problem to the MVC problem. In Section 3, we present a polynomial-time algorithm which finds a minimum sufficient set problem in tree networks. In Section 4, we present a polynomial-time algorithm which finds a minimum sufficient set problem in tree-connected rings. In Section 5, we present a polynomial-time algorithm which finds a minimum sufficient set problem in tree of rings. Finally Section 6 summarizes this paper.

2. Preliminaries

A WDM network in this paper is a bi-directed graph $G = (V, E)$: one for which $(u, v) \in E$ if and only if $(v, u) \in E$. The skeleton of the network G, denoted by $G_s = (V, E')$, is the undirected graph obtained from G by replacing each bi-directed pair of edges with a single undirected edge. By partial abuse of terminology, we will say a set is sufficient in G_s if and only if it is sufficient in G. A vertex v is referred to as a *branching node* if its degree in G_s is greater than 2, a *relay node* if its degree in G_s is equal to 2, or a *leaf node* if its degree in G_s is equal to 1. We will assume that G_s is connected and contains at least one branching node, since otherwise G_s is either a path or a cycle, and the minimum sufficient set can be solved trivially. We say that a node of a path P is an *internal node* in this path if it is not one of the two endpoints.

From the graph G_s, we construct another undirected multigraph $G_c = (V_c, E_c)$, referred to as the *contraction* of the graph G_s, as follows: V_c consists of all branching nodes in G_s. For any two branching nodes u and v, (u, v) is an edge in E_c if and only if there exists a path in G_s between u and v such that all internal nodes in this path are relay nodes. Note that G_c may have self-loops, which we retain as part of the graph. The following lemma establishes the connection between the minimum sufficient set in G and the MVC in G_c.

Lemma 1. [8] *Any MVC of G_c is also a minimum sufficient set of G.*

As a consequence of Lemma 1, it is sufficient to obtain a MVC of the contraction of G_c in order to find a minimum sufficient set in a graph G. In general, the MVC problem is NP-complete and it has a 2-approximation algorithm. But it is fixed-parameter tractable: whether a graph has a VC of size at most k can be decided with time $O(f(k)p(n))$ (see e.g. [7]) where, p is a polynomial function. If the graphs are planar, a polynomial-time approximation scheme exists [5,6]. However, as will show in this paper, when the graph is a tree or a tree-connected rings, a MVC can be found in polynomial time.

3. Minimum sufficient set in trees

In this section, we consider the WDM networks whose underlying topologies are trees. It is obvious that the contraction of any tree is also a tree with the additional property that all internal nodes have degrees of at least three. Thus according to Lemma 1 the minimum sufficient set can be obtained by finding a MVC in the contraction tree. In the next we will present a general polynomial-time algorithm to find MVCs in trees.

We call an *internal* node of a tree to be a *leaf-root* if one of its neighbors is a leaf node. If a tree has no leaf-root then it must be a single edge and its MVC consists of an arbitrary node of this edge. If a tree contains a leaf-root, then the next lemma indicates that there is a MVC which contains this leaf-root.

Lemma 2. *Let G be a tree with a leaf root u. Then G has a MVC which contains u.*

Proof. We prove the lemma by contradiction. Assume the lemma were not true. Let C be a MVC of G which contains the *most* number of leaf-roots of G, but does not contain u. Let v be any leaf node that is a neighbor of u. Clearly, v must be in C, for otherwise the edge (u, v) would not be covered by C. Consider

$$C' = (C - \{v\}) \cup \{u\}.$$

Then C' is also a MVC of G, and contains one more leaf-roots of G than C. This contradicts the selection of C. Thus the lemma is true. □

Let G be any tree and C be any MVC of G which contains a leaf-root u of G. Let G' be the graph obtained from G by

Table 1
Recursive algorithm to find a MVC of a tree

```
Algorithm: MVC_Tree
Input: a tree G;
Output: a MVC of G;
begin
  if G is a single edge
    return an arbitrary node of the edge;
  else
    add a leaf-root of G to the VC of G;
    obtain the residue graph G' of G;
    find the MVC of all componets of G' recursively;
    add them to the VC of G;
end
```

removing u and all its incident edges. Then G' is a forest and it is referred to as the *residue* of G. Let $C' = C \backslash \{u\}$. Then for each component tree T' of G', $C' \cap T'$ is also a MVC of T'. Thus we can apply Lemma 2 to all component trees of G' recursively. Based on this observation, we have the following recursive algorithm to find a MVC of a tree described in Table 1.

The algorithm **MVC_Tree** can be implemented efficiently. The details are omitted over here.

4. Minimum sufficient set in tree-connected rings

The tree-connected rings, illustrated in Fig. 1, is an interconnection topology widely used in the telecommunications industry. In this topology, each node is within a ring and these rings are interconnected via a tree-like topology. It is easy to see that the contraction of any tree-connected rings is also a tree-connected rings. As the MVC of the contraction graph provides an optimal sufficient set of the original graph, we will provide a polynomial-time algorithm that finds a MVC in an arbitrary tree-connected rings.

It is well-known that there are at least two leaf-nodes in any tree. Similarly, one can show that in any tree-connected rings, there exist at least two rings in which all nodes have degree of two except one whose degree is three. Such rings are referred to as *leaf-rings*. The only node in a leaf-ring whose degree is three is called a *bridging-node*. Suppose that a leaf-ring contains m nodes. Then any MVC contains at least $\lceil m/2 \rceil$ nodes in this leaf-ring. In the next we show that there is always a MVC, which contains the bridging node of any leaf ring.

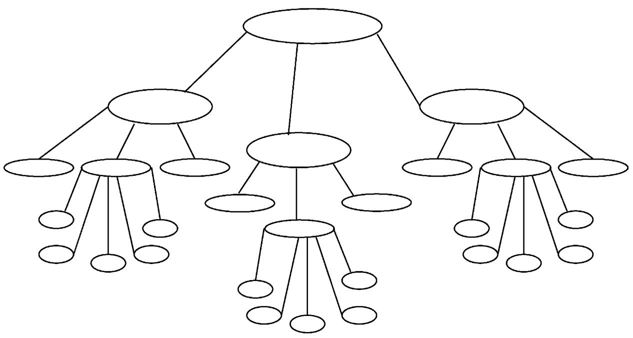

Fig. 1. An example of tree-connected rings.

Lemma 3. *Let G be a tree-connected rings. Then G has a MVC which contains all bridging-nodes of all leaf-rings in G.*

Proof. We prove the lemma by contradiction. Assume the lemma were not true. Let C be a MVC of G which contains the *most* number of bridging-nodes of G, and let u be any bridging-nodes of G that is not in C. Let R be the leaf-ring containing the bridging-node u. Then R cannot be a self-loop, for otherwise u must be in any VC of G and thus in C too. So R contains $m \geq 2$ nodes, say $u = v_1, v_2, \ldots, v_m$ in the clockwise order. Clearly C must contain at least $\lceil m/2 \rceil$ nodes in R, i.e.

$$|C \cap R| \geq \left\lceil \frac{m}{2} \right\rceil.$$

Consider

$$C' = (C - R) \cup \{v_i : 1 \leq i \leq m, i \text{ is odd}\}.$$

Then C' is also a VC of G, and

$$|C'| = |C - R| + \left\lceil \frac{m}{2} \right\rceil = |C| - |C \cap R| + \left\lceil \frac{m}{2} \right\rceil \leq |C|.$$

Thus C' is also a MVC of G. On the other hand, C' contains one more bridging-nodes of G than C. This contradicts the selection of C. Thus the lemma is true. □

The proof of Lemma 3 suggests a MVC of a leaf-ring. Suppose that a leaf-ring R contains m nodes, say $v_1, v_2, \ldots, v_m$ in the clockwise order in which v_1 is its bridging node. The set of vertices

$$\{v_i : 1 \leq i \leq m, i \text{ is odd}\}$$

is called a *canonical* VC of the leaf-ring R. Then Lemma 3 indicates that for any leaf-ring R, there is a MVC C of G which contains its canonical VC. Let $G–R$ denote the graph obtained from G by removing all node in R and their incident edges. Let $C–R$ denote the vertex set obtained from C by removing the canonical VC of R. Then $G–R$ is also a tree-connected rings and $C–R$ is a MVC of G'. On the other hand, the union of the canonical VC of R and any MVC of $G–R$ is also a MVC of G. Based on this observation, we have the following incremental algorithm to find the MVC of a tree-connected rings listed in Table 2.

The algorithm **MVC_Tree_Rings** removes one leaf-ring at each incremental step and add its canonical VC. With carefully selected data structures and implementation, its run-time can be linear in the network size. The details are omitted over here.

Table 2
Algorithm to find a MVC of a tree-connected rings

Algorithm: MVC_Tree_Connected_Rings
Input: *a tree-connected rings G;*
Output: *a MVC of G;*
begin
  $\mathcal{C} = \emptyset$; *// the VC of G*
  while *G is not empty*
    find a leaf-ring R in G;
    add the canonical VC of R to $\mathcal{C}$*;*
    $G = G - R$;
  output $\mathcal{C}$;
end

From the algorithm **MVC_Tree_Rings**, we can explicitly count the cardinality of any MVC of a tree-connected rings.

Theorem 4. *Let $G = (V, E)$ be a tree-connected rings, and k be the number of odd-sized rings in G. Then the cardinality of any MVC of G is $(|V| + k)/2$.*

Proof. Let $R_1, R_2, \ldots, R_l$ be the component rings in G. According to our above algorithm, each ring will eventually become a leaf-ring and its canonical VC will be added to the MVC of G. Note that the canonical VCs of different component rings are disjoint. Thereby, the cardinality of any MVC of G is

$$\sum_{i=1}^{l} \left\lceil \frac{|R_i|}{2} \right\rceil = \sum_{i=1}^{l} \frac{|R_i|}{2} + \frac{k}{2} = \frac{1}{2}\sum_{i=1}^{l} |R_i| + \frac{k}{2} = \frac{|V|}{2} + \frac{k}{2} = \frac{|V| + k}{2}.$$

□ As a corollary of Theorem 4, the cardinality of any minimum sufficient set is at least half of the number of branching-nodes. In order to reduce the number of converters needed, the topology should be carefully designed. For an example, we can choose the topology such that the number of branching nodes in each component ring to be even.

5. Minimum sufficient set in trees of rings

A tree of rings, also known as a *cactus*, is a connected graph whose biconnected components are all rings (Fig. 2), i.e. a pair of vertices have two vertex-disjoint paths if and only if they are within the same ring. From any tree of rings, a tree, referred to its *underlying tree*, can be constructed in the following manner. The vertex set of the underlying tree consists of the original vertices in the tree of rings with

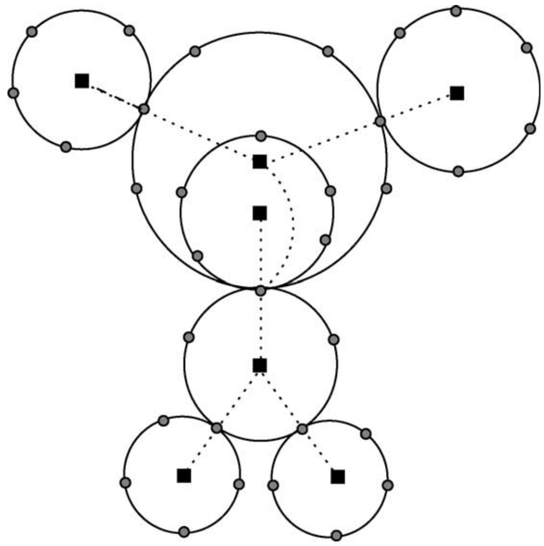

Fig. 2. The tree of rings depicted in solid arcs, and its underlying tree depicted in dashed lines with black squares representing the centers of the rings.

degree (even) at least four and a number of new vertices (the black squares in Fig. 2), each of which corresponds to a unique ring in the tree of rings (imagine the new node as the center of the ring). The edge in the underlying tree exists only between an original vertex and a new vertex, and the edge exists if the original vertex is the inside the ring represented by the new vertex (see the dashed lines in Fig. 2).

The contraction of any tree of rings is also a tree of rings with the property that the nodal degrees are all even numbers greater than two. In addition, the contraction has at least two rings which are self-loops. These rings are also referred to as leaf-rings. Note that any VC of the contraction must contain the unique node in each loop. However, after removing all edges incident to the vertices selected as part of the VC, the residue graph may no longer be a tree of rings, and it may even be disconnected. However, each connected component can be treated as subgraph of a tree of rings, which can be obtained by choosing a subset of rings and remove a link from each of these rings. We refer to each such component as a *sub-tree of rings*. In a sub-tree of rings, we can similarly define leaf-roots as in trees and leaf-rings as in tree-connected rings. A *leaf-root* is an internal node with at least one neighbor as a leaf. A *leaf-ring* is a ring which contains at most one node with nodal degree more than two. For each leaf-ring, one can define its *canonical* VC as for tree-connected rings. By the similar argument, we can show that if a sub-tree of rings has non-empty leaf-roots, then it has a MVC that contains all leaf-roots. If a sub-tree of rings has no leaf-roots, then it must contains at least two leaf-rings. Pick any leaf-ring. Then the sub-tree of rings has a MVC that contains the canonical VC of the picked leaf-ring. Notice that a tree

Table 3
Recursive algorithm to find a MVC of a tree of rings

```
Algorithm: MVC_Sub-tree_of_Rings
Input: a subtree of rings G;
Output: a MVC of G;
begin
  if G is a ring,
    return an arbitrary MVC of the ring;
  if G is a single edge,
    return an arbitrary node of the edge;
  if G has a leaf-root,
    add the leaf-root of G to the VC of G;
  else
    find a leaf-ring of G;
    add the canonical VC of the leaf ring to the VC of G;
  obtain the residue graph G';
  find the MVC of all components of G' recursively;;
  add them to the VC of G;
end
```

of rings can also be treated as a sub-tree of rings. So we propose the following recursive algorithm for MVC of sub-tree of rings as describe in Table 3.

6. Summary

The minimum sufficient set problem is in general NP-complete and is as hard as the MVC problem. However, the underlying topologies of most WDM networks in the telecommunications industry are built around trees, rings, tree-connected rings, and tree of rings. For these topologies, this paper showed that both minimum sufficient problem and the MVC problem can be solved in polynomial time. Efficient algorithms have been provided for the MVC problem in these topologies which in turn are used to solve the minimum sufficient set problem in these topologies.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswani, B. Schieber, M. Sudan, Efficient Routing and Scheduling Algorithms for Optical Networks, Proceedings of the 5th Annual ACM-SAM Symposium on Discrete Algorithms, 1994, pp. 412–423.

[2] Y. Aumann, Y. Rabani, Improved Bounds for All Optical Routing, Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, 1995, pp. 567–576, see also pp. 22–24.

[3] R. Ballart, Y.C. Ching, SONET: now it's the standard optical network, IEEE Communications Magazine (1989) 8–15.

[4] J. Babcock, SONET: a practical perspective, Business Communications Review (1990) 59–63.

[5] B. Baker, Approximation Algorithms for NP-complete Problems on Planar Graphs, Proceedings of 24th IEEE Symposium on Foundations of Computer Science, 1983.

[6] R. Bar-Yehuda, S. Even, On Approximating a Vertex Cover for Planar Graphs, Proceedings of 14th ACM Symposium on Theory of Computing, 1982.

[7] R. Downey, M. Fellows, in: P. Clote, J. Remmel (Eds.), Parametrized Computational Feasibility, in Feasible Mathematics II, Birkhauser, 1994.

[8] J. Kleinberg, A. Kumar, Wavelength Conversion in Optical Networks, Proceedings of 10th ACM-SIAM Symposium on Discrete Algorithms, 1999.

[9] V. Kumar, E. Schwabe, Improved Access to Optical Bandwidth, Proceedings of 8th ACM-SIAM Symposium on Discrete Algorithms, 1997.

[10] M. Mihail, C. Kaklamanis, S. Rao, Efficient Access to Optical Bandwidth, Proceedings of 36th IEEE Symposium on Foundations of Computer Science, 1995.

[11] B. Mukherjee, Optical Communication Networks, McGraw-Hill, New York, 1997.

[12] D. Nesset, T. Kelly, D. Marcenac, All-optical wavelength conversion using SOA nonlinearities, IEEE Communications Magazine (1998) 56–61.

[13] Y. Rabani, Path-coloring on the Mesh, Proceedings of 37th IEEE Symposium on Foundations of Computer Science, 1996.

[14] G.R. Ritchie, SONET lays the roadbed for broadband networks, Networking Management (1990).

[15] P. Raghavan, E. Upfal, Efficient All-optical Routing, Proceedings of 26th ACM Symposium on Theory of Computing, 1994, pp. 134–143.

[16] P. Wilfong, P. Winkler, Ring Routing and Wavelength Translation, Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.