Routing and Wavelength Assignment in Generalized WDM Tree Networks of Bounded Degree*

Stratis Ioannidis[1], Christos Nomikos[2], Aris Pagourtzis[3], and Stathis Zachos[3,4]

[1] Department of Computer Science, University of Toronto
[2] Department of Computer Science, University of Ioannina
[3] School of Elec. & Comp. Engineering, National Technical University of Athens
[4] CIS Department, Brooklyn College, Cuny
`stratis@cs.utoronto.ca`
`cnomikos@cs.uoi.gr`
`{pagour, zachos}@cs.ntua.gr`

Abstract. The increasing popularity of all-optical networks has led to extensive research on the routing and wavelength assignment problem, also termed as the *Routing and Path Coloring problem* (RPC). Here we present a polynomial time algorithm that solves RPC exactly in *generalized tree networks* of bounded degree. This new topology is of practical interest since it models tree-like backbone networks connecting bounded-size LANs of any form. Tree-like backbone structure is very common in practice and bounded size LANs is a reasonable assumption, since LANs are by nature networks unable to sustain a large number of hosts.

Keywords: *routing and path coloring, wavelength assignment, WDM networks, optical networking.*

1 Introduction

All-Optical Networks. Optical fiber technology evolves rapidly and will eventually lead to networks consisting solely of optical connections. In such networks, which are called *all-optical*, signal multiplexing is achieved through the WDM (*wavelength division multiplexing*) technique; a different wavelength is assigned to each signal, thus enabling multiple signals to be transmitted through the same fiber.

The bottleneck in current optical connections lies in the conversion of electrical signals to optical ones and vice versa. Consequently, it appears that all-optical networks should minimize such conversions. To that purpose, messages that are routed through the network should be processed as little as possible. One way to achieve this is to maintain the same wavelength throughout the network for each signal transmitted from one host to another. In other words, when two (possibly

---

* Research supported in part by "Pythagoras" grant of the Ministry of Education of Greece, co-funded by the European Social Fund (75%) and National Resources (25%) — Operational Programme for Education and Initial Vocational Training (EPEAEK II).

P. Bozanis and E.N. Houstis (Eds.): PCI 2005, LNCS 3746, pp. 57–67, 2005.

non-adjacent) hosts communicate with each other through a connection, optical signals realizing this connection use only one wavelength.

The Routing and Path Coloring Problem. All-optical networks can be modeled as graphs and *connection requests* between two hosts can be represented by pairs of nodes. In the case of *full-duplex* communication the requests, hence also the pairs of nodes, are actually unordered. Such requests are *routed* into actual connections through appropriate paths connecting the two nodes. Each of these connections uses the same wavelength for all transmissions, so wavelength assignment can be modeled as coloring of the corresponding path. The fact that two signals can be multiplexed in a fiber only if different wavelengths are assigned to them imposes restrictions on possible colorings of the path set. If we assume that adjacent hosts are connected through a single fiber link, a coloring shall be feasible only if *overlapping* paths (i.e. sharing at least one edge) are assigned different colors.

The *Routing and Path Coloring problem* (RPC) is stated as follows: Given a graph G and a set of connection requests R (unordered pairs of nodes), find a routing of R, i.e. a set of corresponding paths P, and a feasible coloring of paths in P using the least possible number of colors. Note that the minimum number of colors among all possible routings and colorings is sought. It also makes sense to consider the problem where the routing is pre-determined and only an optimal coloring is sought; this is known as the *Path Coloring problem* (PC): Given a graph G and a set of paths P, find a feasible coloring of paths in P using the least possible number of colors.

We are interested in simple paths only, since any routing and coloring that uses non-simple paths can be transformed to one with only simple paths and at most the same cost (number of colors needed). Hence, in acyclic topologies there is a unique possible routing and RPC coincides with PC. Note also that paths are not necessarily distinct, i.e. two paths may pass through identical edges and nodes. This captures the possibility of more than one connections between two nodes, all using the same fiber links.

Related Work. Both RPC and PC have been studied for a variety of topologies, such as chains, rings, trees, trees of rings, grids, to name only a few.

As mentioned above, the two problems coincide in acyclic topologies and can be solved exactly within polynomial time in chains [31] and in bounded degree trees [26,10]. Furthermore, PC in stars is equivalent to edge coloring [14] in general multigraphs and thus is NP-hard [16] and approximable within a factor of $\frac{4}{3}$ [15,25]. The same approximation factor can be achieved in unbounded degree trees [26,10].

PC in rings, also known as the *Circular-Arc Coloring problem*, was proved to be NP hard by Garey et al. [13]; Shih and Hsu [33] presented an approximation algorithm with ratio $\frac{5}{3}$ and Karapetian [17] developed a $\frac{3}{2}$-approximation algorithm. A randomized algorithm that achieves a 1.37 asymptotic approximation factor with high probability, in instances where the optimum solution is large, was given by V. Kumar [20]. Results combining the above with approximation algorithms for PC in trees yield approximation results for trees of rings [26,24,8]. RPC in rings is also NP-hard [10,4]; simple algorithms with approximation ratio 2 were presented by Raghavan and Upfal [32] (undirected problem) and by Mihail et al. [24] (directed problem); A randomized algorithm for RPC in rings with asymptotic approximation ratio 1.38 was given by V. Kumar [19] and was later improved by Cheng [5].

PC in undirected trees can be approximated within a ratio of 1.1 (asymptotically). This is due to the fact that PC in undirected trees is equivalent to edge coloring of multigraphs via approximation preserving reductions [14]; hence the 1.1-approximation algorithm of Nishizeki and Kashiwagi [25] for edge coloring of multigraphs yields an algorithm for PC in undirected trees with the same approximation ratio. In directed trees, the best known algorithm is a 5/3-approximation (using at most $5L/3$ colors) due to Erlebach et al. [11].

S. R. Kumar et al. [18] proved that PC in trees can be solved exactly in polynomial time provided that the degree of the tree is bounded. They did this by showing that RPC in constant size graphs (with unbounded number of requests) is in P; to this end, they employed an integer programming technique. On the other hand, Nomikos [26] proved that PC is non-approximable in meshes (unless P=NP), and thus in arbitrary graphs as well. Extensive work has been done for PC and RPC in (unbounded) directed tree topologies [24,11,18,10].

A related problem is the MAXIMUM ROUTING AND PATH COLORING PROBLEM (MAXRPC), where the goal is to route and color as many requests as possible using only the available colors; the variation where the routing is pre-determined is called MAXIMUM PATH COLORING PROBLEM (MAXPC). MAXPC in chains, also known as the "k-coloring of intervals" problem, can be solved exactly [3]. Nomikos, Pagourtzis and Zachos [29] have proposed a 3/2-approximation algorithm for MAXRPC in undirected rings and a 11/7-approximation algorithm for the directed case; they have also given a 3/2-approximation algorithm for MAXPC in rings, both for the undirected and the directed case [30]. For trees, a 1.58 approximation for the undirected case was presented by Wan and Liu [34], while for the directed case a 2.22-approximation is due to Erlebach and Jansen [9].

Variants of PC and RPC, where multiple fibers are allowed, have been extensively studied in recent years [28,35,22,23,27,12]; in these papers, constant ratio approximation algorithms are presented for various basic topologies such as chains, rings, stars and trees. In [1,2] they give lower and upper bounds on the approximability of multiple-fiber problems for general topologies. Other related work includes traffic grooming; in this approach one can combine low speed traffic components onto high speed channels in order to minimize network cost. Traffic grooming in path, star and tree networks is NP-complete [7]; the same holds for ring networks [6].

Generalized Trees. A generalized tree G is constructed from a set S of connected graphs (LANs) and a tree T as follows: Every node v of T is replaced by a LAN V from S. Every edge (u, v) of T is replaced by an edge (u', v'), where u' is any node of LAN U and v' is any node of LAN V (U replaces u and V replaces

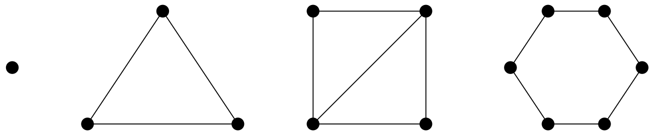

Fig. 1. A finite set of graphs (LANs)

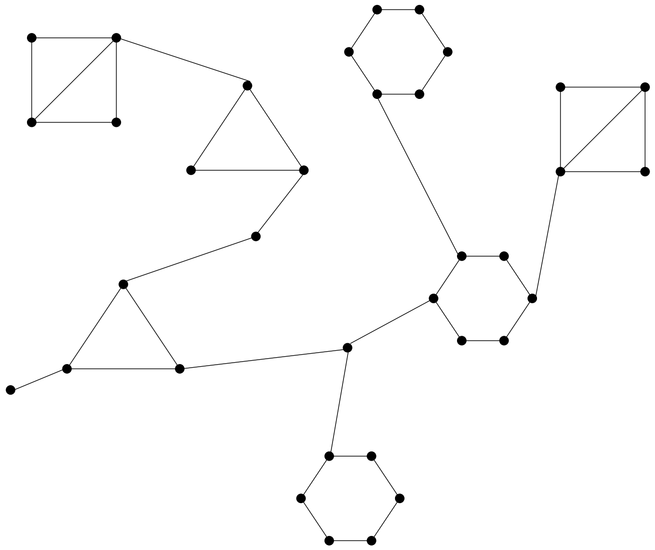

Fig. 2. A generalized tree with LANs from the set shown in Figure 1

v). We call (u', v') a *bridge.* An example of a set of LANs and a generalized tree are shown in figures 1 and 2 respectively.

We define the degree of a generalized tree to be the degree of the underlying tree T. We shall focus on generalized trees with the following two properties:

1. The degree of the generalized tree is bounded by a fixed constant D.
2. The size (number of nodes) of the LANs is bounded by a fixed constant B.

Definition 1. *We call graphs with the above properties* g-trees of degree $\leq D$ and LAN size $\leq B$*, or simply* g-trees*.*

The path coloring problem (PC) in generalized trees is NP-hard, since a tree is a special case of a generalized tree, and PC for trees is known to be NP-hard [14]. In contrast, we will show that PC in g-trees can be solved exactly in polynomial time. This is quite surprising, considering that PC and RPC are NP-hard in very simple topologies (e.g. in rings and in unbounded degree trees) end even non-approximable in simple topologies like meshes.

The g-tree topology is of particular practical interest because it captures a wide class of existing networks: tree-like backbone structure and bounded size LANs are reasonable assumptions, since LANs are by nature networks unable to sustain a large number of hosts; besides, the degree of the underlying tree is often small, therefore it makes sense to consider it bounded.

2 A Polynomial-Time Algorithm

Theorem 1. *For any $D, B > 0$ there exists a polynomial-time algorithm that solves PC for any given g-tree of degree $\leq D$ and LAN size $\leq B$.*

We will show the above theorem by presenting such an algorithm. The algorithm has two ingredients: a *recursive coloring* technique and a subroutine for coloring g-stars (g-trees where the underlying graph is a star).

2.1 Recursive Coloring

Let (G, P) be an instance of PC in g-trees, that is G is a g-tree and P is a set of paths on G. In the following we adapt ideas from [32] to show how to color paths in P recursively.

If G contains more than one LAN, it also has a bridge e, i.e. an edge of the underlying tree T, joining two LANs. We break G into two g-trees G_1 and G_2 as shown in Figure 3. From the set of paths P two sets of paths P_1 and P_2 are constructed:

$$P_1 = \{q \mid q = p \cap G_1,\ p \in P\}$$
$$P_2 = \{q \mid q = p \cap G_2,\ p \in P\}$$

That is, every path that passes through edge e is split into two subpaths, whereas paths not passing through e are contained either in P_1 or in P_2. Such a split is shown in Figure 4.

This partitioning gives two new instances of PC in g-trees, namely (G_1, P_1) and (G_2, P_2). Given colorings of these two instances we can produce a common coloring of the initial instance (G, P) using a number of $\max(c_1, c_2)$ colors, where c_i is the number of colors used for instance (G_i, P_i). Paths in P_1, P_2 that were created by splitting a path in P must be assigned the same color. This can be easily done in polynomial time by a color permutation. Such a permutation is feasible because all paths that pass through edge e overlap and thus all belong to different color classes in each of the colorings of (G_1, P_1) and (G_2, P_2). Obviously, the number of colors used is $\max(c_1, c_2)$. Moreover, if the colorings (G_1, P_1) and (G_2, P_2) are optimal, so is the coloring of (G, P), otherwise an optimal coloring would induce a coloring on one of the instances with fewer colors, a contradiction.

All paths in G can be colored recursively by repeated application of the above splitting technique. In order to fully define the recursion scheme, we need to be able to color the paths at the basis of the recursion, i.e. for only one LAN, with at most D additional edges (rays). We call such a graph a *g-star*.

In the following we show that a g-star can be colored optimally in polynomial time.

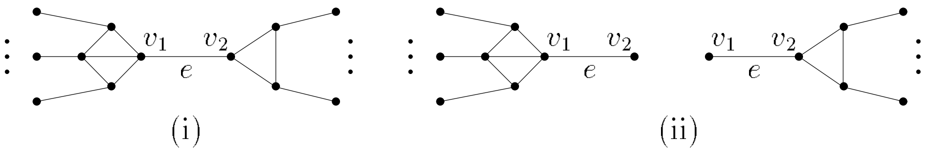

Fig. 3. Dividing a g-tree G (i) into G_1 and G_2 (ii) over bridge e

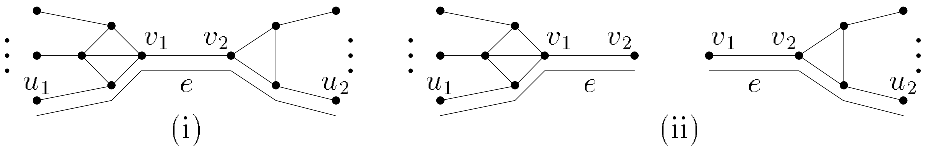

Fig. 4. Dividing a path p on G (i) into a path on G_1 and a path on G_2 (ii) over bridge e

2.2 Coloring a g-Star

Consider a g-star H consisting of a LAN V of size at most B with at most D rays. (Note: the total number of nodes is at most $B + D$.) We show that there exists an algorithm that colors a set of paths P in H in time polynomial in the size of P.

Two paths are called *distinct* if they differ in at least one edge. Since the size of H is bounded by $B + D$ (a constant), the number of (simple) distinct paths between any two nodes of H is also bounded by a constant c. A rather rough estimation for this constant is $c = (D + 1)^2(B + 1)!/2$ which can be shown as follows: each path on H can be seen as a permutation of the nodes of LAN V — possibly together with a starting (ending) node outside V attached to the first (resp. last) node of the permutation; there are $(B + 1)!$ such permutations (we add a dummy "finishing" node in order to take into account paths that contain fewer than B nodes in V as well), and each permutation, together with its possible endpoints outside V, corresponds to at most $(D + 1)^2$ paths (again using a dummy node to capture cases where paths start or end inside V); since paths are undirected they are encountered twice; dividing by two we obtain our estimation. Let Q be the set of all distinct paths in H. We denote with $Q_1, Q_2, \ldots, Q_l$ all the non-empty subsets of Q that contain *edge-disjoint* paths (i.e. paths that do not have common edges). Obviously, $l \leq 2^c$ and thus l is also a constant, independent of the size of the given input. Note that, actually, constant l is usually much smaller because most subsets of Q contain paths which are not edge-disjoint.

Consider an instance (H, P) of PC. A coloring of paths in P is merely a partitioning of P into subsets of edge-disjoint paths (color classes); therefore, each color class corresponds to a subset Q_i. As we have already pointed out, P may contain more than one "copies" of a path, i.e. two or more non-distinct paths. Two copies of the same path belong to different color classes in any coloring of P; hence it may happen that two color classes are described by the same subset Q_i. In such a case, it does not make any difference to the cardinality of the solution (i.e. the number of colors used), which copy of a path is used in which color class; an interchange between two copies in a feasible solution yields yet another feasible solution of the same cardinality. Therefore, a coloring of P can be described in general as a multiset of Q_i sets.

Assume that k_i is the multiplicity of Q_i in a solution. Note that k_i is bounded by p_i, where p_i is the minimum multiplicity of a path in P among paths contained in Q_i. Let $p = \max_{1 \leq i \leq l} p_i$; clearly, p is bounded by the maximum multiplicity of any path in P. An optimal solution can be found by exhaustive search: out of all possible values of $k_1, k_2, \ldots, k_l$, where $0 \leq k_i \leq p_i$, we choose as feasible solutions the corresponding multisets that constitute a partitioning of P. Out of these, the optimal solution is the one with the least cardinality, i.e. the one that minimizes the quantity $\sum k_i$.

Checking whether a multiset is a partitioning of P can be done in $O(|P|)$ steps, and the number of all possible multisets is $O((p+1)^l)$. Thus, for any g-star H, PC can be solved in time $O(|P|p^l)$ (also $O(|P|^{l+1})$, where l is a constant depending only on B and D and not on the size of the input.

Note that an alternative approach would be to use [18]'s idea of formulating the problem as an integer program and solving it using Lenstra's algorithm [21]. However, this does not seem to improve the time complexity, since it also involves explicit generation of all path matchings (sets of edge-disjoint paths) before solving the IP.

2.3 Complexity of the g-tree Algorithm

The number of g-stars in a g-tree $G = (V, E)$ is at most $|V|$ and at least $|V|/B$, therefore it is $\Theta(|V|)$, and the number of paths in each g-star is at most $|P|$. Hence, the complexity of coloring the g-stars is $O(|V||P|^{l+1})$. Color permutations (recoloring) can be done in $O(|V||P|)$ time in total [26]. Thus, there exists an algorithm that solves PC for g-trees of degree $\leq D$ and LAN size $\leq B$ in $O(|V||P|^{l+1})$, where l is a constant depending only on D and B.

3 Routing and Path Coloring

We next explain how to extend the above algorithm to an algorithm for the *Routing and Path Coloring problem (RPC)* on g-trees. An instance of RPC consists of a g-tree G and a set R of *connection requests*, i.e. pairs of nodes. A feasible solution is a routing of the connection requests into a set of paths P, and a coloring of this set so that overlapping paths are not assigned the same color. The goal is to minimize the number of colors used.

We now show how to adapt the algorithm for PC in g-trees so as to obtain a polynomial-time algorithm that solves RPC in g-trees exactly. We break G into

G_1 and G_2 as before (with bridge e) and introduce a request-splitting technique as follows: a request between a node in G_1 and a node in G_2 is split into two requests, each one containing a node from the original pair and an endpoint of bridge e. Since bridge $e = (v_1, v_2)$ is contained in any path connecting a node w_1 of G_1 with a node w_2 of G_2, any routing of a request (w_1, w_2) can be seen as a routing of a request from w_1 to bridge v_2 and a routing of a request from bridge v_1 to w_2 (see Figure 5). After color assignment, the two solutions can be combined using the permutation technique as before; optimality of the partial solutions guarantees the optimality of the combined solution.

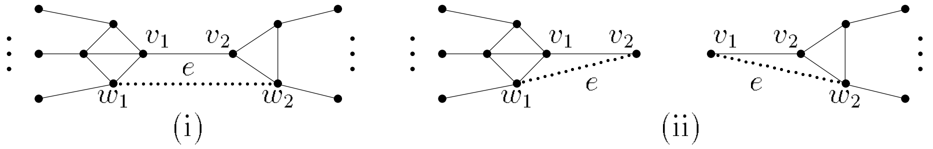

Fig. 5. Dividing a request on G (i) into a request on G_1 and a request on G_2 (ii) over bridge e

The basis of the recursion is practically the same as before: each multiset of Q_i sets is a feasible solution if it defines a set of paths P which implements a routing of requests in R. The optimal solution will be again a feasible solution of minimum cardinality. Therefore, the following is true:

Theorem 2. *For any $D, B > 0$ there exists a polynomial-time algorithm that solves RPC for any given g-tree of degree $\leq D$ and LAN size $\leq B$.*

4 Algorithm Improvement

The performance of the above algorithm can be improved substantially by reducing l, i.e. the number of Q_i sets. To this end it suffices to observe that only maximal sets of edge-disjoint paths are necessary: for any two sets Q_i, Q_j of edge-disjoint paths with $Q_i \subset Q_j$, Q_j can replace Q_i in any solution that contains Q_i, without affecting the cost of the solution. In the modified algorithm, a multiset of Q_i sets is a feasible solution if it defines a superset of P rather than P itself. This modification will ensure that the algorithm yields an optimal solution. This follows from two properties: First, for any multiset of Q_i sets which is a minimum cardinality partition of P (i.e. is an optimal solution) there exists a multiset with the same cardinality that includes only maximal Q_i sets and defines a superset of P. Second, from every multiset of maximal Q_i sets which defines a superset of P one can construct a partition of P with cardinality at most the same.

The above technique may render the algorithm practicable enough, since the number of maximal Q_i sets can be relatively small compared to the total number of Q_i sets. More specifically, for each maximal Q_i, all non-empty subsets of Q_i are present in the initial collection; all these subsets are now represented by Q_i alone.

5 Conclusions

We have studied PC in g-trees and proved that it can be solved exactly by a polynomial time algorithm. This result is of practical interest, considering the wide variety of networks that can be modeled by g-trees. We also showed how to extend this result to RPC in g-trees, that is, for the case in which the topology is not acyclic and thus routing is not trivial.

An interesting question remains, namely whether there exist similar or more generic topologies than g-trees for which PC and RPC are solvable in polynomial time. Our technique does not seem to apply immediately to the case where requests and paths are directed; however, similar topologies may exist for which a polynomial solution can be obtained for the directed case as well.

Future research may also address such questions in variations of PC such as considering weighted graphs, multi-fiber networks, as well as maximization problems and traffic grooming.

Future research may also address such questions for PC/RPC variants. For example, it would be interesting to consider PC and RPC in weighted graphs or in multifiber networks, as well as to study maximization versions of the problem and traffic grooming.

References

1. M. Andrews and L. Zhang. Wavelength Assignment in Optical Networks with Fixed Fiber Capacity. In *Proc. ICALP 2004*, pp. 134–145.
2. M. Andrews and L. Zhang. Bounds on fiber minimization in optical networks with fixed fiber capacity. In *Proc. INFOCOM 2005*.
3. M. C. Carlisle and E. L. Lloyd. On the k-coloring of intervals. *Discrete Applied Mathematics*, 59:225–235, 1995.
4. T. Carpenter, S. Cosares, and I. Saniee. Demand routing and slotting on ring networks. Technical Report 97-02, 28, 1997.
5. C. C. T. Cheng. A new approximation algorithm for the demand routing and slotting problem with unit demands on rings. In *Proceedings RANDOM-APPROX'99*, Lecture Notes in Computer Science, 1671:209–220, 1999.
6. A. L. Chiu and E. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *Journal of Lightwave Technology*, 18(1):2–12, 2000.
7. R. Dutta, S. Huang, and G. N. Rouskas. Traffic grooming in path, star, and tree networks: complexity, bounds, and algorithms. In *Proc. of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 298–299. ACM Press, 2003.
8. T. Erlebach. *Scheduling Connections in Fast Networks.* PhD thesis, Technische Universität München, 1999.

9. T. Erlebach and K. Jansen. Maximizing the number of connections in optical tree networks. In *Proceedings of the 9th Annual International Symposium on Algorithms and Computation (ISAAC'98)*, LNCS 1533, pages 179–188, 1998.
10. T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255(1–2):33–50, 2001.
11. T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, and P. Persiano. Optimal wavelength routing in directed fiber trees. *Theoretical Computer Science*, 221:119–137, 1999.
12. T. Erlebach, A. Pagourtzis, K. Potika, and S. Stefanakos. Resource allocation problems in multifiber WDM tree networks. In *29th Workshop on Graph Theoretic Concepts in Computer Science (WG 2003)*, Elspeet, Netherlands, June 2003.
13. M. R. Garey, D. S. Johnson, G. L. Miller, and C. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal of Algorithms and Discrete Mathematics*, 1(2):216 - 227, 1980.
14. M. C. Golumbic and R. E. Jamison. The edge intersection graphs of paths in a tree. *J. Comb. Theory Series B*, 38(1):8–22, February 1985.
15. D. S. Hochbaum, T. Nishizeki, and D. B. Shmoys. A better than"best possible" algorithm to edge color multigraphs. *Journal of Algorithms*, 7:79–104, 1986
16. I. Holyer. The NP-completeness of edge coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981.
17. I. A. Karapetian. On the coloring of circular arc graphs. Docladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic, 70(5):306–311, 1980, in Russian.
18. S. R. Kumar, R. Panigrahy, A. Russel, and R. Sundaram. A note on optical routing on trees. *Information Processing Letters*, 62(6):295–300, 1997.
19. V. Kumar. *Bandwidth Allocation Algorithms for All-Optical Networks.* PhD thesis, Northwestern University, Evanston, Illinois, 1997.
20. V. Kumar. Approximating circular arc colouring and bandwidth allocation in all-optical ring networks. In *Proceedings of Approximation Algorithms for Combinatorial Optimization Problems (APPROX '98)*, Aalborg, Denmark, 1998.
21. H.W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.* 8 (1983), pp. 538–548.
22. G. Li and R. Simha. On the wavelength assignment problem in multifiber WDM star and ring networks. *IEEE/ACM Transactions on Networking*, 9(1):60–68, 2001.
23. L. Margara and J. Simon. Wavelength assignment problem on all-optical networks with k fibres per link. In *Proceedings of the 27th International Colloquium on Automata, Languages, and Programming (ICALP 2000)*, LNCS 1853, pages 768–779, 2000.
24. M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. In *Proc. IEEE Symp. Foundations of Computer Science*, pages 548–557, Milwaukee, Wisconsin, 1995.
25. T. Nishizeki and K. Kashiwagi. On the 1.1 edge-coloring of multigraphs. *SIAM J. Disc. Math.*, 3(3):391–410, August 1990.
26. C. Nomikos. *Path Coloring in Graphs.* PhD thesis, National Technical University of Athens, Athens, 1997.
27. C. Nomikos, A. Pagourtzis, K. Potika, and S. Zachos. Fiber Cost Reduction and Wavelength Minimization in Multifiber WDM Networks. In *Proc. NETWORKING 2004*, LNCS 3042, pp. 150–161, 2004.
28. C. Nomikos, A. Pagourtzis, and S. Zachos. Routing and path multicoloring. *Information Processing Letters*, 80(5):249–256, December 2001.

29. C. Nomikos, A. Pagourtzis, and S. Zachos. Minimizing request blocking in all-optical rings. In *Proc. INFOCOM 2003*.
30. C. Nomikos, A. Pagourtzis, and S. Zachos. Satisfying a maximum number of pre-routed requests in all-optical rings. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 42(1):55–63, 2003.
31. S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37:21–25, 1991.
32. P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proceedings of the Annual ACM Symposium on Theory of Computing STOC '94*, pages 134–143, 1994.
33. W. K. Shih and W. L. Hsu. An approximation algorithm for coloring circular-arc graphs. In *SIAM Conference on Discrete Mathematics*, 1990.
34. P.-J. Wan and L. Liu. Maximal throughput in wavelength-routed optical networks. In *Multichannel Optical Networks: Theory and Practice*, volume 46 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 15–26. AMS, 1998.
35. P. Winkler and L. Zhang. Wavelength assignment and generalized interval graph coloring. In *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 830–831, Baltimore, MD, January 2003.