In: Proceedings of the 9th Annual International Symposium on Algorithms and Computation ISAAC'98, Lecture Notes in Computer Science 1533, pp. 179–188, 1998. ©Springer-Verlag

Maximizing the Number of Connections in Optical Tree Networks

Thomas Erlebach*,[1] and Klaus Jansen**,[2]

[1] TU München, 80290 München, Germany, erlebach@in.tum.de
[2] IDSIA Lugano, Corso Elvezia 36, 6900 Lugano, Switzerland, klaus@idsia.ch

Abstract. In optical networks with wavelength division multiplexing (WDM), multiple connections can share a link if they are transmitted on different wavelengths. We study the problem of satisfying a maximum number of connection requests in a directed tree network if only a limited number W of wavelengths are available. In optical networks without wavelength converters this is the *maximum path coloring* (MaxPC) problem, in networks with full wavelength conversion this is the *maximum path packing* (MaxPP) problem. MaxPC and MaxPP are shown to be polynomial-time solvable to optimality if the tree has height one or if both W and the degree of the tree are bounded by a constant. If either W or the degree of the tree is not bounded by a constant, MaxPC and MaxPP are proved $\mathcal{NP}$-hard. Polynomial-time approximation algorithms with performance ratio $5/3 + \varepsilon$ for arbitrarily small ε are presented for the case $W = 1$, in which MaxPC and MaxPP are equivalent. For arbitrary W, a 2-approximation for MaxPP in arbitrary trees, a 1.58-approximation for MaxPC in trees of bounded degree, and a 2.22-approximation for MaxPC in arbitrary trees are obtained.

1 Introduction

All-optical communication networks are the technology of choice for satisfying the ever-growing demands for telecommunication bandwidth. Data is transmitted through optical fiber at the rate of gigabits-per-second, and WDM (wavelength-division multiplexing) allows several connections to use a link simultaneously if the signals are transmitted on different wavelengths. In all-optical networks with switches without wavelength conversion capabilities, a connection must use the same wavelength on the whole path from transmitter to receiver. If W wavelengths are available, a set of connections can be established if each connection is assigned a transmitter-receiver path and one of the W wavelengths such that connections sharing a directed link receive different wavelengths. Another model of optical networks employs switches with wavelength conversion. A connection can use different wavelengths on different segments of its transmitter-receiver path. If all network switches have full wavelength conversion capabilities, a set of connections can be established if they are assigned transmitter-receiver paths such that no directed link is used by more connections than the number of available wavelengths.

* supported by DFG contract SFB 342 TP A7
** supported partially by the EU ESPRIT LTR Project NO. 20244 (ALCOM-IT)

As the number of distinct available wavelengths is small in practice, it is not always the case that all requests in a given set of connection requests can be established simultaneously under the constraints mentioned above. In such a scenario, the network provider might decide to reject some requests and to maximize the number of accepted requests. We investigate the complexity and approximability of this optimization problem for the case that the topology of the network is that of a *directed tree*, i.e., the graph obtained from an undirected tree by replacing each undirected edge by two directed edges with opposite directions.

1.1 Preliminaries

A *connection request* in a directed tree T is given by a sender-receiver pair (u, w) and corresponds to the directed path from u to w in T. We will refer to requests as paths in the remainder of this paper. Two paths *intersect* if they share a directed edge of T. For a given set P of paths, the *load* $L(e)$ of a directed edge e of T is the number of paths in P using edge e, and L denotes the maximum load among all edges of T. A W-coloring of a given set of paths is an assignment of colors (wavelengths) to the paths using at most W colors such that intersecting paths receive different colors.

For given directed tree T, set P of paths in T, and number W of available colors, the *maximum path coloring* problem (MaxPC) is to compute a subset $P' \subseteq P$ and a W-coloring of P' such that $|P'|$ is maximized, and the *maximum path packing* problem (MaxPP) is to compute a subset $P' \subseteq P$ with maximum load W such that $|P'|$ is maximized. For a given instance of MaxPC or MaxPP, we denote by P^* an arbitrary optimum solution. MaxPC models optical networks without wavelength converters, while MaxPP models the availability of full wavelength conversion.

By Δ we denote the maximum outdegree of all nodes in the given directed tree T. We assume that an arbitrary node r of T has been designated the root of T. For $v \neq r$, $p(v)$ denotes the parent of v. The *level* of a node is its distance (number of edges) from r. For a pair (u, w) of nodes in T, we denote by $\mathrm{lca}(u, w)$ the unique *least common ancestor* (lca) of u and w, i.e., the node with smallest level among all nodes on the path from u to w. The one or two edges on a path (u, w) that are incident to $\mathrm{lca}(u, w)$ are called the *top edges* of the path (u, w). A subtree of T *contains* a path (u, w) if $\mathrm{lca}(u, w)$ is a node of the subtree.

A polynomial-time algorithm is a ρ-approximation algorithm for MaxPC or MaxPP if it always outputs a set P' whose cardinality is at least a $(1/\rho)$-fraction of the cardinality of an optimum solution. An algorithm that computes an optimum solution in polynomial time is called an *exact* algorithm.

1.2 Related Work

Previous work has focused on the *path coloring problem*, where the goal is to assign wavelengths to all given connection requests while minimizing the number of different wavelengths used. For undirected trees, a $(3/2)$-approximation algorithm was given in [10] and improved to an asymptotic 1.1-approximation in [2]. For directed trees, the best known algorithm colors a given set of directed paths with maximum load L using at most $\lceil (5/3)L \rceil$ colors [8, 5, 7, 6]. While the path coloring problem is relevant when a provider designs a network in order to meet the given demands or when the network has enough capacity for satisfying all given requests, MaxPC and MaxPP apply to the case where an existing network has insufficient capacity and the goal is to maximize the number of accepted requests.

In chain networks, MaxPC and MaxPP are equivalent and can both be solved optimally in polynomial time by finding a maximum W-colorable subgraph in the conflict graph, which is an interval graph in this case [12]. For undirected trees, the exact algorithm for integral multicommodity flow with unit edge capacities from [3] (see below) gives an exact algorithm for MaxPC and MaxPP with $W = 1$, and using the reduction from Sect. 4 a 1.58-approximation for MaxPC with arbitrary W is obtained. The same results can be derived for ring networks [11]. For ring networks with predetermined routing of the given requests, a $3/2$-approximation algorithm for MaxPC was obtained in [9]. A variant of the MaxPC problem is considered in [1]; see Sect. 4 for more details.

MaxPP is closely related to the integral multicommodity flow problem. Integral multicommodity flow and multicut have been studied for undirected trees in [3]. For a given set of source-sink pairs (commodities) in an undirected tree with edge capacities, the integral multicommodity flow problem is to maximize the sum of the flows of the commodities constrained by the given edge capacities. Exact algorithms for integral multicommodity flow are obtained in [3] for trees of height one with arbitrary edge capacities and for arbitrary trees with unit edge capacities. For trees with edge capacities 1 or 2 the problem is proved $\mathcal{NP}$-hard and MAX SNP-hard. For trees with arbitrary edge capacities, a 2-approximation algorithm is given. The main differences between the multicommodity flow problem considered in [3] and the MaxPP problem studied in the present paper are that we investigate directed instead of undirected trees and that no commodity can have a flow greater than 1 in our setting.

1.3 Results

We determine the complexity of MaxPP and MaxPC in directed trees under various restrictions and give approximation algorithms with small constant approximation ratios for the variants that are $\mathcal{NP}$-hard. We are not aware of any previous results regarding MaxPP and MaxPC in directed trees. Our complexity results are listed in this section without giving proofs; details can be found in the full paper available from the authors.

MaxPP and MaxPC are equivalent if the given tree has height one and can be solved in polynomial time using an algorithm for capacitated b-matching [4, pp. 257–259]. This algorithm extends to the weighted version of MaxPC and MaxPP, where each path p has associated benefit $b(p)$ and the goal is to maximize the total benefit of accepted paths. MaxPP and MaxPC can also be solved optimally in polynomial time using a bottom-up computation if the maximum degree Δ of the given tree network and the number W of available wavelengths are bounded by a constant. This algorithm extends to the weighted version of MaxPC and MaxPP as well, and also to variants where the set of available wavelengths can vary from link to link or where wavelength converters with limited conversion are allowed. (Furthermore, variants of the algorithm give exact algorithms for integral multicommodity flow in directed or undirected trees of bounded degree, if the edge capacities are bounded by a constant.) If either Δ or W can be arbitrarily large, MaxPP and MaxPC become $\mathcal{NP}$-hard. More precisely, both problems are $\mathcal{NP}$-hard for $W = 1$ and arbitrary degree, and for arbitrary W and degree bounded by 3 (i.e., binary trees).

For MaxPP with arbitrary W, we adapt the algorithm from [3] and obtain a 2-approximation (Sect. 2). If $W = 1$, i.e., only one wavelength is available, then MaxPC and MaxPP are equivalent to finding a maximum cardinality subset of edge-disjoint paths, and we give, as our main result, a family of polynomial-time approximation algorithms with approximation ratio $5/3 + \varepsilon$ for this case, where ε can be chosen arbitrarily small (Sect. 3). For MaxPC with arbitrary W, we obtain a 2.22-approximation for trees of arbitrary degree and a 1.58-approximation for trees whose degree is bounded by a constant (Sect. 4).

2 Approximating MaxPP

The algorithm is as follows. Initially, set $P' = \emptyset$. Then process all nodes of the tree in order of non-increasing levels. When processing node v, consider the paths whose lca is v in arbitrary order. Insert each such path in P' if this does not increase the maximum load of P' above W. In the end, output P'. (Note that this algorithm for MaxPP works also if the number of available wavelengths is different on different links.)

Theorem 1. *The algorithm is a* 2*-approximation algorithm for MaxPP, i.e., it outputs a subset* P' *of* P *with maximum load at most* W *and with cardinality at least* $|P^*|/2$*.*

Proof. First, we observe that the approximation algorithm from [3] works also for directed trees. The only further difference between the multicommodity flow problem in [3] and the MaxPP problem is that, with the MaxPP problem, no commodity can have flow greater than 1. But our greedy algorithm for MaxPP in a directed tree T with edge capacity W behaves like the algorithm from [3] in a slightly extended tree T': for each path (commodity) p from a node u to a node w, add two new nodes u_p and w_p, add two unit capacity edges (u_p, u) and (w, w_p), and replace p by a path from u_p to w_p. The multicommodity flow problem in the resulting tree T' is equivalent to the MaxPP problem in the original tree, and our greedy algorithm produces the same solution as the 2-approximation algorithm from [3] on this instance. □

3 A Family of Approximation Algorithms for $W = 1$

The algorithm from the previous section achieves approximation ratio 2 also for $W = 1$. The main idea that leads to an improvement in this special case is to consider all paths with the same lca simultaneously instead of one by one. This way we obtain, for any fixed $\varepsilon > 0$, a polynomial-time $(5/3 + \varepsilon)$-approximation algorithm.

Let P_v denote the subset of all paths $(u, w) \in P$ with $\mathrm{lca}(u, w) = v$ that do not intersect any of the paths that have been accepted by the algorithm at a previous node and that do not use any edges that have been *reserved* or *fixed* by the algorithm (see below). We assume without loss of generality that we have $u \neq v \neq w$ for all paths $(u, w) \in P_v$.

During a first pass, the algorithm processes the nodes of T in order of non-increasing levels. When the algorithm processes node v, it tries to determine for the paths in P_v

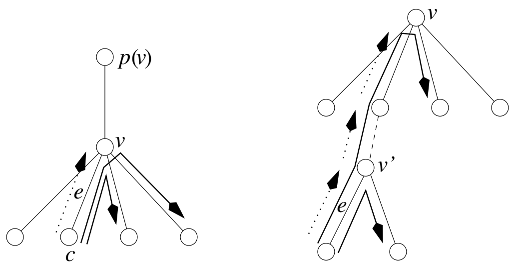

Fig. 1. Possible configurations of deferred paths

whether they should be included in P' (these paths are called *accepted*) or not (these paths are called *rejected*). But the algorithm can not always make this decision right away. In some cases it leaves some paths in an intermediate state and resolves them later on. The possibilities for paths in such intermediate states are *undetermined paths*, *groups of deferred paths*, *groups of exclusive paths*, and *groups of* 2*-exclusive paths*. We refer to undetermined paths, groups of exclusive paths, and groups of 2-exclusive paths (but not groups of deferred paths) as *unresolved* paths.

If all paths in P_v use the same two top edges, accepting one of them might cause the algorithm to miss the chance of accepting two paths with an lca of smaller level later on. Instead, the algorithm picks one of the paths in P_v and makes it an *undetermined path* in this case.

Sometimes the algorithm decides to accept one of several intersecting paths, but it defers the decision which one of them to accept. The intersecting paths are called a *group of deferred paths* (see Fig. 1), and some edges (indicated by dotted arrows in Fig. 1) are marked as *reserved*. The motivation for introducing groups of deferred paths is as follows: first, the reserved edges block at most one path with lca of smaller level that could be accepted in an optimum solution; second, no matter which paths with lca of smaller level not intersecting a reserved edge are accepted by the algorithm later on, there is still at least one deferred path that can be accepted in a second pass that proceeds top-down.

A *group of exclusive paths* is sketched in Fig. 2(a). Such a group consists of a *lower path* p and a *higher path* q with lca of smaller level that intersects p. At most one of the two paths can be accepted, but the algorithm can not afford to pick the wrong one. It only marks the top edge of p that is intersected by q as *fixed* (indicated by a dotted arrow in Fig. 2). Groups of exclusive paths have the following property.

Property (E): *As long as at most one path touching* v *but not using the fixed edge is accepted at a later node, either* p *or* q *can still be accepted. Only when two paths touching* v *are accepted at a later node, they may block both* p *and* q *from being accepted (see Fig. 2(b)).*

The last types of unresolved paths are sketched in Fig. 2(c) and (d). These *groups of* 2*-exclusive paths* consist of a set of four paths at most two of which can be accepted and have the following property.

Property (2E): *If at most one path touching* v *but not using a fixed edge is accepted at a later node, two paths from the group of* 2*-exclusive paths can still be accepted. If two*

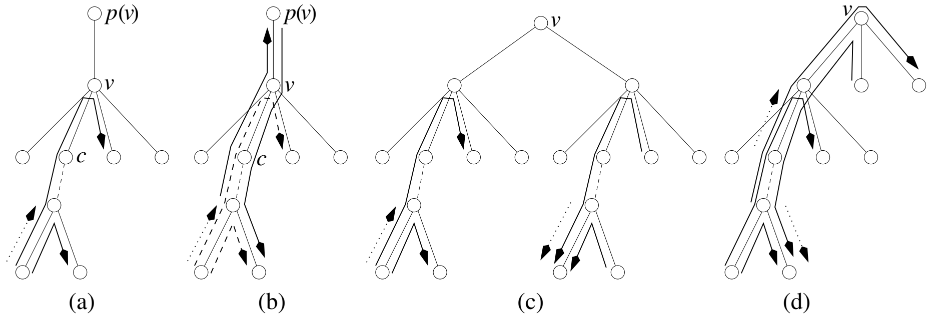

Fig. 2. (a) Possible configuration of exclusive paths; (b) situation in which both exclusive paths are blocked; (c) and (d) groups of 2-exclusive paths

paths touching v but not using a fixed edge are accepted at a later node, at least one path from the group of 2-exclusive paths can still be accepted.

When the algorithm has finished processing a node v, the subtree rooted at v will contain at most one of the following: one undetermined path, or one group of exclusive paths, or one group of 2-exclusive paths. All other paths in the subtree are accepted, rejected, or member of a group of deferred paths.

3.1 Invariants

In the next subsection we will sketch how the algorithm proceeds during the first pass. At the same time, we will show that the approximation ratio achieved by the algorithm is $5/3 + \varepsilon$. For establishing this, it can be proved by induction that the following invariant can be maintained. This invariant holds before the first node of T is processed, and it holds again each time an additional node of T has been processed.

Let A be the set of all paths that have already been accepted by the algorithm. Let F be the set of all paths in P whose lca has not yet been processed and which are not blocked by any of the accepted paths, by reserved edges of deferred paths, or by fixed edges. Let d be the number of groups of deferred paths that are contained in processed subtrees. Let U be the set of all undetermined paths. Let X be the union of all groups of exclusive paths and groups of 2-exclusive paths. Then there is a set $O \subseteq F \cup U \cup X$ of edge-disjoint paths satisfying the following conditions:

Condition 1: $|P^*| - |O| \leq (5/3 + \varepsilon)(|A| + d)$

Condition 2: *For every group of exclusive paths, O contains one path from that group; for every group of 2-exclusive paths, O contains two paths from that group.*

O represents a set of paths that could still be accepted by the algorithm and that has the following property: if the algorithm accepts at least a $1/(5/3 + \varepsilon)$-fraction of the paths in O (in addition to the paths it has already accepted), its output is a $(5/3 + \varepsilon)$-approximation of the optimum solution.

The invariant is satisfied initially with $A = \emptyset$, $d = 0$, $F = P$, $U = \emptyset$, $X = \emptyset$, and $O = P^*$. In order to prove that the invariant can be maintained, it suffices to show how the set O that establishes the invariant before node v is processed can be manipulated so as to satisfy the invariant also after node v is processed. In particular, some paths must be replaced in O or removed from O in order to satisfy $O \subseteq F \cup U \cup X$ and to keep O a set of edge-disjoint paths after v is processed, and it must be shown that the number of paths removed from O is at most $(5/3 + \varepsilon)(a_v + d_v)$ if the algorithm accepts a_v additional paths and creates d_v new groups of deferred paths while processing v (thus satisfying Condition 1). Condition 2 must only be considered explicitly when a new group of exclusive paths or group of 2-exclusive paths is created.

If the invariant is satisfied after the root node is processed, we have $F = \emptyset$, $O \subseteq U \cup X$, and $|P^*| - |O| \leq (5/3 + \varepsilon)(|A| + d)$. By accepting the undetermined path (if any), accepting an arbitrary path from the group of exclusive paths (if any), and accepting two arbitrary edge-disjoint paths from the group of 2-exclusive paths (if any), the algorithm accepts $|O|$ additional paths, and the resulting set A satisfies $|P^*| \leq (5/3 + \varepsilon)(|A| + d)$. In the second pass, the algorithm processes the nodes of the tree in order of non-decreasing levels and accepts g additional paths at each node v that is the lca of g groups of deferred paths. Then the algorithm outputs the set of all accepted paths. As the algorithm accepts d additional paths, one from each group of deferred paths, in the second pass, this establishes our main theorem.

Theorem 2. *For every fixed $\varepsilon > 0$, there is a polynomial-time approximation algorithm for MaxPC and MaxPP with one available color (wavelength) having approximation ratio $5/3 + \varepsilon$.*

3.2 The First Pass

Recall that $P_v \subseteq P$ is the set of all paths with lca v that do not intersect any previously accepted path nor any fixed or reserved edge. Let U_v be the set of undetermined paths contained in subtrees rooted at children of v. Let X_v be the union of groups of exclusive paths and groups of 2-exclusive paths contained in subtrees rooted at children of v. We sketch how the algorithm processes node v and determines which of the paths in $P_v \cup U_v \cup X_v$ should be accepted, rejected, deferred, or left in an unresolved state.

Note that for a given set of paths with lca v the problem of determining a maximum subset of edge-disjoint paths is equivalent to finding a maximum matching in a bipartite graph [8] and can thus be done in polynomial time. Furthermore, we use the following property of bipartite graphs: for $s = 1$ or $s = 2$, the fact that a maximum matching in a bipartite graph G has cardinality s implies that there are s vertices in G such that every edge is incident to at least one of these s vertices. (The property holds for arbitrary s and is known as the König theorem.)

Let k be the number of children of v that have an undetermined path in their subtree, let ℓ be the number of children that have a group of exclusive paths, and let m be the number of children that have a group of 2-exclusive paths. We use the expression *subtrees with exclusive paths* to refer to all subtrees of v with either a group of exclusive paths or with a group of 2-exclusive paths. Before v is processed, the invariant implies that there is a set $O \subseteq F \cup U \cup X$ satisfying Condition 1 and 2. In each single case of the following case analysis, it must be shown how a set O' can be obtained from O such that Condition 1 and 2 are satisfied for O' after v is processed.

Case 1: $k + \ell + m \leq \max\{3, 2/\varepsilon\}$. The algorithm can try out all combinations of accepting or rejecting unresolved paths in the subtrees rooted at children of v: for undetermined paths there are two possibilities (accepting or rejecting the path), for groups of exclusive paths there are two possibilities (accepting the lower path or accepting the higher path), and for groups of 2-exclusive paths there are at most four relevant possibilities of accepting two edge-disjoint paths of the group (see Fig. 2). Hence, the number of possible combinations is bounded from above by $2^{k+\ell}4^m = O(1)$.

For each of these combinations, the algorithm can compute a maximum number of edge-disjoint paths in P_v not intersecting any of the (tentatively) accepted paths from $U_v \cup X_v$. Let s be the maximum, over all combinations, of the number of tentatively accepted paths from $U_v \cup X_v$ plus the number of maximum edge-disjoint paths in P_v. If $s = 0$, we have $k = \ell = m = 0$ and $P_v = \emptyset$, and the algorithm proceeds with the next node. Otherwise, we distinguish the following cases.

If $s \geq 3$, the algorithm accepts the s paths and rejects all other paths from $P_v \cup U_v \cup X_v$. As s is the maximum number of edge-disjoint paths in $P_v \cup U_v \cup X_v$, O can contain at most s paths from $P_v \cup U_v \cup X_v$. Furthermore, O can contain at most two paths from F using the edges $(v, p(v))$ or $(p(v), v)$, and these are the only two further paths in O that could possibly be blocked by the s paths accepted by the algorithm. Hence, a valid set O' can be obtained from O by deleting at most $s + 2$ paths. As $s + 2 \leq (5/3)s$, the invariant is maintained. In the cases $s = 1$ and $s = 2$, a huge number of subcases for k, ℓ and m such that $k+\ell+2m \leq 1$ resp. $k+\ell+2m \leq 2$ is distinguished. For each of these subcases, a number of configurations of paths in P_v with respect to the paths in $X_v \cup U_v$ are considered. Each such subcase and configuration can be recognized in polynomial time, and it can be shown that one of the following actions can satisfy the invariant: (1) The algorithm creates a new undetermined path or a new group of exclusive paths. A valid set O' can be derived from O by replacing or inserting one path, if necessary. (2) The algorithm creates a new group of deferred paths. In this case, a valid set O' can be derived from O by deleting at most one path. (3) The algorithm accepts paths and/or creates groups of deferred paths such that $a_v + d_v = 2$. In this case, a valid set O' can be derived from O by deleting at most 3 paths. (4) The algorithm creates a group of 2-exclusive paths from some paths in $P_v \cup X_v \cup U_v$. In this case, a valid set O' can be derived from O by replacing or inserting at most two paths. All details are omitted in this extended abstract due to space limitations.

Case 2: $k + \ell + m > \max\{3, 2/\varepsilon\}$. The algorithm calculates four candidate sets $S_1,\ldots,S_4$ of edge-disjoint paths from $P_v \cup U_v \cup X_v$ and chooses the largest of them. For obtaining S_2 and S_4, we employ a method of removing r paths from an arbitrary set S of edge-disjoint paths in P_v such that $\ell + 2m$ exclusive paths from X_v can be accepted in addition to the paths remaining in S. The details of the method and a proof that $r \leq (|S| + \ell + m)/3$ are omitted. S_1 is obtained as the union of all k undetermined paths and a maximum number s_1 of edge-disjoint paths from P_v not intersecting any undetermined path, and as many additional edge-disjoint paths from the $\ell + m$ subtrees with exclusive paths as possible. We have $|S_1| \geq k + s_1 + m$, because S_1 contains k undetermined paths and at least m paths from groups of 2-exclusive paths in X_v due to Property (2E). S_2 is obtained from S_1 by removing r of the s_1 paths in $S_1 \cap P_v$ from S_1 until $\ell + 2m$ exclusive paths can be accepted. S_2 contains $\ell + 2m$ exclusive paths, and only $r \leq (s_1 + \ell + m)/3$ of the s_1 paths in $S_1 \cap P_v$ were removed to obtain S_2. As S_2 still contains the k undetermined paths, we have $|S_2| \geq k + m + (2/3)(s_1 + \ell + m)$. S_3 is obtained by taking a maximum number s_3 of edge-disjoint paths from P_v and as many additional edge-disjoint paths from the $\ell + m$ subtrees with exclusive paths and the k subtrees with undetermined paths as possible. We have $|S_3| \geq s_3 + m$, because S_3 contains at least m paths from groups of 2-exclusive paths in X_v due to Property (2E). S_4 is obtained from S_3 by removing r of the s_3 paths in $S_3 \cap P_v$ from S_3 until $\ell + 2m$ exclusive paths can be accepted. Since $r \leq (s_3 + \ell + m)/3$, we have $|S_4| \geq m + (2/3)(s_3 + \ell + m)$. We claim that the number of paths in $O_v = O \cap (P_v \cup U_v \cup X_v)$ is at most $s_1 + (s_3 - s_1)/2 + k + \ell + 2m$. With this upper bound on $|O_v|$ and the lower bounds on the cardinalities of the four sets S_i, it can be proved that at least one of the sets S_i satisfies $|O_v| + 2 \leq (5/3 + \varepsilon)|S_i|$. At most $|O_v| + 2$ paths must be removed from O in order to obtain a valid set O'.

4 Approximating MaxPC for Arbitrary *W*

In order to obtain an approximation algorithm for MaxPC with arbitrary number W of available wavelengths from an algorithm for $W = 1$, we employ a technique from [1]. The approximation algorithm A for arbitrary number W of wavelengths is obtained from an approximation algorithm A_1 for one wavelength by running W copies of A_1 and giving as input to the i-th copy the set of paths that have not been accepted by the first $i - 1$ copies of A_1. The output of A is the union of the W sets of paths output by the copies of A_1, and the paths in the i-th set are assigned colored i.

In [1] it is shown that the algorithm A obtained in this way has approximation ratio at most $\rho + 1$ if A_1 has approximation ratio ρ, even if different wavelengths are associated with different network topologies. There the technique is used to obtain randomized on-line algorithms with logarithmic competitive ratio for networks shaped as rooted forests. For identical networks, which we have in our application, the approximation ratio achieved by A can even be bounded by $1/(1 - (1 - 1/(\rho W))^W)$, which is smaller than $1/(1 - \mathrm{e}^{-1/\rho})$ for all W. This bound is mentioned in a preliminary draft of the journal version of [1], which was kindly supplied to the authors by Adi Rosén. The bound can be proved easily by using the fact that, if A has selected p_k paths after running k copies of A_1, there is still a set of at least $(|P^*| - p_k)/W$ edge-disjoint paths among the remaining paths (this follows from a pigeonhole argument), and the next copy of A_1 accepts at least a $(1/\rho)$-fraction of them. As we have obtained an exact algorithm for MaxPC with $W = 1$ in bounded degree trees and $(5/3 + \varepsilon)$-approximation algorithms for MaxPC with $W = 1$ in arbitrary trees, by employing the above technique we obtain approximation algorithms with ratio $1/(1 - 1/e) \approx 1.58$ for arbitrary W in bounded degree trees and with ratio ≈ 2.22 for arbitrary W in arbitrary trees.

5 Open Problems

It is an interesting open question whether the approximation ratios of our algorithms for directed trees can be improved. Other promising directions for future research include approximation algorithms for the weighted MaxPP and MaxPC problems in arbitrary trees and for the MaxPC problem with different sets of available wavelengths on different links. In addition, it would be very interesting to see whether techniques we used in the $(5/3 + \varepsilon)$-approximation for MaxPC and MaxPP with $W = 1$ can lead to improved approximation algorithms for the integral multicommodity flow problem in trees in general or for special cases thereof; for this problem, the best known approximation is still the 2-approximation from [3].

Acknowledgments

We are grateful to Stefano Leonardi for pointing out the reduction from MaxPC with arbitrary number of wavelengths to MaxPC with one wavelength, and to Adi Rosén for informing us about the improved analysis for the ratio obtained by this reduction in the case of identical networks for all wavelengths.

References

1. B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosén. On-line competitive algorithms for call admission in optical networks. In *Proceedings of the 4th Annual European Symposium on Algorithms ESA '96*, LNCS 1136, pages 431–444. Springer-Verlag, 1996.
2. T. Erlebach and K. Jansen. Scheduling of virtual connections in fast networks. In *Proceedings of the 4th Parallel Systems and Algorithms Workshop PASA '96*, pages 13–32. World Scientific Publishing, 1997.
3. N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming, ICALP '93*, LNCS 700, pages 64–75. Springer-Verlag, 1993.
4. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, Berlin, 1988.
5. C. Kaklamanis and P. Persiano. Efficient wavelength routing on directed fiber trees. In *Proceedings of the 4th Annual European Symposium on Algorithms ESA '96*, LNCS 1136, pages 460–470. Springer-Verlag, 1996.
6. C. Kaklamanis, P. Persiano, T. Erlebach, and K. Jansen. Constrained bipartite edge coloring with applications to wavelength routing. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming ICALP '97*, LNCS 1256, pages 493–504. Springer-Verlag, 1997.
7. V. Kumar and E. J. Schwabe. Improved access to optical bandwidth in trees. In *Proceedings of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms SODA '97*, pages 437–444, 1997.
8. M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 548–557, 1995.
9. C. Nomikos and S. Zachos. Coloring a maximum number of paths in a graph. Presented at the *Workshop on Algorithmic Aspects of Communication (July 11–12, 1997, Bologna, Italy)*.
10. P. Raghavan and E. Upfal. Efficient routing in all-optical networks. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing STOC '94*, pages 134–143, New York, 1994. ACM SIGACT, ACM Press.
11. P.-J. Wan and L. Liu. Maximal throughput in optical networks, 1998. Presented at the *DIMACS Workshop on Optical Networks (March 16–19, 1998)*.
12. M. Yannakakis and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. *Inf. Process. Lett.*, 24(2):133–137, January 1987.