Routing and Wavelength Assignment in Optical Networks using Bin Packing Based Algorithms

Nina Skorin-Kapov
Department of Telecommunications, Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia,
E-mail: nina.skorin-kapov@fer.hr

October 28, 2005

Abstract

This paper addresses the problem of routing and wavelength assignment (RWA) of static lightpath requests in wavelength routed optical networks. The objective is to minimize the number of wavelengths used. This problem has been shown to be NP-complete and several heuristic algorithms have been developed to solve it. We suggest very efficient, yet simple, heuristic algorithms for the RWA problem developed by applying classical bin packing algorithms. The heuristics were tested on a series of large random networks and compared with an efficient existing algorithm for the same problem. Results indicate that the proposed algorithms yield solutions significantly superior in quality, not only with respect to the number of wavelength used, but also with respect to the physical length of the established lightpaths. Comparison with lower bounds shows that the proposed heuristics obtain optimal or near optimal solutions in many cases.

Keywords: OR in telecommunications, routing and wavelength assignment, bin packing, optical networks

1 Introduction

The large potential bandwidth in wavelength-division multiplexed (WDM) optical networks makes WDM technology of crucial importance for satisfying the ever increasing capacity requirements in telecommunication networks. Wavelength routed WDM networks can exploit the large bandwidth of optical fibers by dividing it among different wavelengths. These networks are equipped with configurable WDM nodes which enable us to set up and tear down all-optical channels, called *lightpaths*, between pairs of nodes. Lightpaths can traverse multiple physical links and essentially create a virtual topology on top of the physical topology. Information sent via a lightpath does not require any opto-electronic conversion at intermediate nodes and, thus, greatly reduces delay. Lightpath requests that determine the virtual topology can be known *a priori* (static), or arrive unexpectedly with random holding times (dynamic).

Given a network and a set of lightpath requests, the *Routing and Wavelength Assignment* (RWA) problem attempts to route each lightpath request, and to assign wavelengths to these routes subject to the following constraints. If no wavelength converters are available, the same wavelength must be assigned along the entire route. This is called the *wavelength continuity constraint*. In addition, lightpaths that share a common physical link cannot be assigned the same wavelength. This is called the *wavelength clash constraint*. The objective of the RWA problem is often to minimize the number of wavelengths used, or to maximize the number of lightpaths successfully set up subject to a limited number of wavelengths. This problem has been shown to be NP-complete [3]. Several variations of the problem and their solutions have been proposed in [10] and [14].

We improve upon solutions proposed for the routing and wavelength assignment of static lightpath requests by efficiently applying bin packing algorithms. Bin packing is a classical NP-hard optimization problem [7] which finds its application in many real world problems, such as truck loading, stock-cutting problems, storage allocation for computer networks, the problem of packing commercials into breaks and many others. However, the potential of this model has not yet been systematically explored in the context of the routing and wavelength assignment problem.

We apply bin packing to develop very efficient - yet simple - heuristic algorithms for the RWA problem with the objective to minimize the number of wavelengths used. We also consider a second objective, which is to minimize the physical lengths of the established lightpaths. The motivation for these objectives is as follows. Minimizing the number of wavelengths is desirable in order to leave more room for future expansion of the virtual topology. Minimizing the physical length of a lightpath, not only in terms of hops, but also in terms of actual distance, is desirable in all WDM networks due to signal degradation and propagation delay. Furthermore, in opaque networks where electronic regeneration is performed at each node, minimizing the physical hop length of individual lightpaths is crucial. Such networks require a transmitter and receiver at the head and tail nodes, respectively, of each physical link included in the lightpath. As a result, longer physical paths dramatically increase the cost of the network.

The algorithms were tested on large random networks and compared with an efficient RWA algorithm presented in [13]. Results indicate that the proposed algorithms obtain solutions which, not only use significantly fewer wavelengths, but which also establish shorter lightpaths. The obtained solutions were also compared with analytical lower bounds. For denser networks, the proposed algorithms obtained optimal or near optimal solutions with respect to both wavelengths and lightpath lengths in many cases. Furthermore, the speed and simplicity of the algorithms make them highly tractable for large networks with many lightpath requests.

The rest of the paper is organized as follows: In Section 2, we informally define the RWA problem, and discuss related work in Section 3. In Section 4 we introduce classical bin packing and suggest heuristic algorithms for the RWA problem. Lower bounds are briefly discussed in Section 5. Numerical results and concluding remarks are given in Sections 6 and 7, respectively.

2 Problem Definition

The physical optical network is modelled as a graph $G = (V, E_p)$, where V is the set of nodes and E_p is the set of physical edges. Edges are assumed to be bidirectional, each representing a pair of optical fibers (i.e. one fiber per direction). Given is a set of lightpath requests $\tau = \{(s_1, d_1), \ldots, (s_n, d_n)\}$, where $s_i, d_i \in V, i = 1, \ldots, n$. Each lightpath request (s_i, d_i) in G is defined by its source node s_i and destination node d_i. The static Routing and Wavelength Assignment problem searches for a set of directed paths $P = \{P_1, \ldots, P_n\}$ in G, each corresponding to one lightpath request, and assigns wavelengths to these paths. Paths P_i and P_j where $i \neq j$, $i, j = 1, \ldots, n$, cannot be assigned the same wavelength if they share a common directed edge. The length[1] of path P_i, $i = 1, \ldots, n$, denoted as $l(P_i)$, can be upper bounded by a value H. The objective is to minimize the number of wavelengths required to successfully route and assign wavelengths to all the lightpath requests in τ. We also consider a second objective which is to minimize the average physical length of the established lightpaths, i.e. $min \frac{\sum_{j=1}^{n} l(P_j)}{n}$.

3 Related Work

Most approaches used to solve the RWA problem in WDM optical networks decompose the problem into two subproblems, routing and wavelength assignment, solved subsequently. A classification of such RWA algorithms can be found in [4]. In [2], the authors use a multicommodity flow formulation and randomized rounding to solve the routing subproblem. Wavelength assignment is solved using graph coloring heuristics. In [8], the authors use local random search for route selection. For each routing scheme, the wavelength assignment problem is solved using a greedy graph coloring algorithm. A generalization of the graph coloring problem, called the partition coloring problem, and its application to routing and wavelength assignment is studied in [12]. In [15], wavelength assignment of previously calculated alternative paths is solved using a tabu search algorithm suggested for partition coloring.

An algorithm which solves the routing and wavelength assignment subproblems simultaneously is suggested in [11]. Here, the authors present an integer formulation and a column

---

[1] Length can be considered in terms of the number of hops or actual distance.

generation technique to help solve it. This approach may not be practical for larger problems. A fast yet effective greedy algorithm based on edge disjoint path (EDP) algorithms is presented in [13]. The algorithm, called *Greedy_EDP_RWA*, creates a partition $\tau_1, \ldots, \tau_k$ of the set of lightpath requests τ. Each element of the partition is composed of a subset of lightpath requests which can be routed on mutually edge disjoint paths in G and, hence, can be assigned the same wavelength. The number of distinct wavelengths needed to successfully perform RWA corresponds to the number of elements in the partition. This algorithm is very simple and fast and yet was shown to outperform the algorithm presented in [2]. The algorithm in [15] was shown to perform the same or slightly better than the multi-start *Greedy_EDP_RWA* algorithm with respect to the number of wavelengths used, after 10 minutes of computational time, for networks with the number of nodes ranging from 14 to 32.

4 Heuristic algorithms for RWA using a bin packing approach

4.1 Bin packing

The bin packing problem is a classical combinatorial optimization problem that has been widely studied in literature. Given is a list of n items of various sizes and identical bins of limited capacity. To solve the bin packing problem, it is necessary to pack these items into the minimum number of bins, without violating the capacity constraints, so that all items are packed. Since this problem is NP-hard [7], a vast array of approximation algorithms have been proposed and studied. Surveys of bin packing algorithms can be found in [6] and [5]. A more recent heuristic algorithm is suggested in [1].

Four well-known classical bin packing algorithms are the First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) algorithms. The FF and BF algorithms are so-called on-line bin packing algorithms, which means that they pack items into bins in random order with no information of subsequent items. Both algorithms label bins in sequential order as new bins are used. The FF algorithm packs each item into the first bin (i.e. the bin with the lowest index) into which it fits. The BF algorithm packs each item into the bin which leaves the least room left over after packing the item.

The FFD and BFD algorithms are two very fast and well known off-line bin packing algorithms. This means that they have information of all the items to be packed *a priori*. Having this information, it seems logical to first place larger items into bins and then fill up the remaining space with smaller items. On the contrary, if all the small items are neatly packed into one bin, there is a great chance that none of the large items will fit into that bin. Moreover, each larger item may need an extra bin of its own leaving a lot of unused space around it and ultimately leading to a larger number of bins used. The FFD and BFD algorithms apply this concept by sorting the given items in nonincreasing order of their corresponding sizes, and then perform packing in the same manner as the FF and BF algorithms, respectively. These algorithms perform significantly better than FF and BF.

4.2 Algorithms for the RWA problem

To apply bin packing to the Routing and Wavelength Assignment problem we must define items, bins, and their corresponding sizes in terms of optical networks. We consider lightpath requests to represent items, while copies of graph G represent bins. Each copy of G, referred to as bin $G_i, i = 1, 2, 3, \ldots$, corresponds to one wavelength. We consider the size of each lightpath $(s_j, d_j) \in \tau$ to be the length of its shortest path SP_j in graph G. However, it is important to note that lightpaths are not necessarily routed on their shortest paths. This measure is used only by the FFD and BFD algorithms in order to sort the 'items' or lightpaths in nonincreasing order of their corresponding sizes.

The capacity of each bin is limited by the edges in G. Namely, two lightpaths routed on the same copy of G cannot traverse any of the same edges due to the *wavelength clash constraint*. To solve the RWA problem, we wish to pack as many items (lightpaths) into a minimum number of bins (copies of G), and therefore minimize the number of wavelengths used. In doing so, we must also take care to satisfy the wavelength continuity and clash constraints. Herein, we propose RWA algorithms, to be referred to as FF_RWA, BF_RWA, FFD_RWA and BFD_RWA, which are respectively based on classical bin packing algorithms FF, BF, FFD and BFD. The FF_RWA algorithm obtains solutions equivalent to those obtained by the *Greedy_EDP_RWA* algorithm suggested in [13], while the remaining algorithms perform significantly better.

4.2.1 *FF_RWA* (***Greedy_EDP_RWA*** [13])

The First Fit bin packing algorithm modified to solve the Routing and Wavelength Assignment problem, referred to as FF_RWA, runs as follows. First, only one copy of G, bin G_1, is created. Higher indexed bins are created as needed. Lightpath requests (s_j, d_j) are selected at random and routed on the lowest indexed copy of G in which there is room. Bin G_i is considered to have room for lightpath (s_j, d_j) if the length of the shortest path from s_j to d_j in G_i, denoted as P_j^i, is less than H. If a lightpath is routed in bin G_i, the lightpath is assigned wavelength i and the edges along path P_j^i are deleted from G_i. If all the edges from bin G_i are deleted, the bin no longer needs to be considered. If no existing bin can accommodate lightpath request (s_j, d_j), a new bin is created.

The FF_RWA algorithm is similar to the *Greedy_EDP_RWA* algorithm suggested in [13]. The difference is in the order in which some steps are executed. Namely, the FF_RWA algorithm routes each lightpath on the first copy of G it fits in. If all the existing bins are full, a new bin is created. The *Greedy_EDP_RWA* algorithm, on the other hand, creates only one copy of G at a time, and then tries to route as many lightpaths as possible on that copy. Due to its basic equivalency with FF_RWA, we will compare the *Greedy_EDP_RWA* from [13] with the rest of the bin packing algorithms proposed in this paper.

```
FF_RWA (FFD_RWA)
Input:
$G = (V, E_p)$;//physical network
$\tau = \{(s_1, d_1), \ldots, (s_n, d_n)\}$; //lightpath requests
$H$; //max physical length of lightpath
Begin:
(ONLY FOR FFD_RWA: Sort and renumerate demands $\tau$ in non-increasing order of their shortest paths, $SP_j$, in $G$ )
$P = \{\}$; //The final paths
Create 1 copy (bin) of $G : G_1$;
$BINS := \{G_1\}$;
while $\tau$ is not empty do
  for $j = 1$ to $|\tau|$ do
    $P_j = \emptyset$;
    for $i = 1$ to $|BINS|$ do
      Find shortest path, $P_j^i$, for lightpath $(s_j, d_j)$ in $G_i$;
      if $l(P_j^i) \leq H$ then
        $P_j = P_j^i$;
        Assign wavelength $i$ to path $P_j$;
        Delete edges in $P_j^i$ from $G_i$;
        $i = |BINS|$;
      end if;
    end for;
    if $P_j = \emptyset$ then
      $New := |BINS| + 1$;
      Create copy of $G$: $G_{New}$;
      $BINS := BINS \cup \{G_{New}\}$;
      Find shortest path, $P_j^{New}$, for lightpath $(s_j, d_j)$ in $G_{New}$;
      $P_j = P_j^{New}$;
      Assign wavelength $New$ to path $P_j$;
      Delete edges in $P_j^{New}$ from $G_{New}$
    end if;
    $P = P \cup P_j$;
    $\tau = \tau \setminus (s_j, d_j)$;
  end for;
end while;
return $P$;
End
```

```
BF_RWA (BFD_RWA)
Input:
$G = (V, E_p)$;//physical network
$\tau = \{(s_1, d_1), \ldots, (s_n, d_n)\}$; //lightpath requests
$H$; //max physical length of lightpath
Begin:
(ONLY FOR BFD_RWA: Sort and renumerate demands $\tau$ in non-increasing order of their shortest paths, $SP_j$, in $G$ )
$P = \{\}$; //The final paths
Create 1 copy (bin) of $G : G_1$;
$BINS := \{G_1\}$;
while $\tau$ is not empty do
  for $j = 1$ to $|\tau|$ do
    $P_j = \emptyset$, $l(P_j) = \infty$;
    $BestBin := 0$;
    for $i = 1$ to $|BINS|$ do
      Find shortest path, $P_j^i$, for lightpath $(s_j, d_j)$ in $G_i$;
      if $l(P_j^i) \leq H$ and $l(P_j^i) < l(P_j)$ then
        $BestBin = i$;
        $P_j = P_j^i$;
        Assign wavelength $i$ to path $P_j$;
      end if;
    end for;
    if $P_j \neq \emptyset$ then
      Delete edges in $P_j^{BestBin}$ from $G_{BestBin}$;
    else
      $New := |BINS| + 1$;
      Create copy of $G$: $G_{New}$;
      $BINS := BINS \cup \{G_{New}\}$;
      Find shortest path, $P_j^{New}$, for lightpath $(s_j, d_j)$ in $G_{New}$;
      $P_j = P_j^{New}$;
      Assign wavelength $New$ to path $P_j$;
      Delete edges in $P_j^{New}$ from $G_{New}$
    end if;
    $P = P \cup P_j$;
    $\tau = \tau \setminus (s_j, d_j)$;
  end for;
end while;
return $P$;
End
```

Figure 1: Pseudocodes of the FF_RWA, BF_RWA, FFD_RWA, and BFD_RWA algorithms.

4.2.2 *BF_RWA*

The Best Fit Routing and Wavelength Assignment algorithm, BF_RWA, routes lightpaths in the bin into which they fit 'best'. The BF_RWA algorithm defines the 'best fit' quite differently than the BF bin packing algorithm. Namely, in classical bin packing, the 'best fitting' bin is considered to be the one in which there remains the least empty space after packing an item. The BF_RWA algorithm, on the other hand, considers the best bin to be the one in which the lightpath can be routed on the shortest path. In other words, if at some point in running the algorithm, there are B bins created, bin G_i, $1 \leq i \leq B$, is considered to be the best bin for lightpath (s_j, d_j) if $l(P_j^i) \leq l(P_j^k)$, for all $k = 1, \ldots, B$, and $k \neq i$. This is not necessarily the overall shortest path, SP_j, since it is possible that none of the existing bins have this path available. If there is no satisfactory path available in any of the B bins (i.e. $l(P_j^i) > H$, for $i = 1, \ldots, B$), a new bin is created.

The motivation for the 'best fit' approach described above, is not only to use less wavelengths, but also to minimize the physical length of the established lightpaths. Of course, we could route each lightpath (s_j, d_j) strictly on its shortest path SP_j, but this would in most cases lead to a larger number of bins, which in turn means using a larger number of wavelengths.

4.2.3 *FFD_RWA*

The First Fit Decreasing Routing and Wavelength Assignment algorithm sorts the lightpath requests in nonincreasing order of the lengths of their shortest paths, SP_j, in G. Lightpaths with shortest paths of the same length are placed in random order. The algorithm then proceeds as FF_RWA.

The motivation for such an approach is as follows. If the connection request with the longest shortest path is considered first, it will be routed in 'empty' bin G_1, i.e. $G_1 = G$. This means the lightpath will not only successfully be routed in G_1, but will be routed on its overall shortest path. After deleting the corresponding edges from bin G_1, the remaining edges can be used to route 'shorter' lightpath requests which are easier to route on alternative routes that are satisfactory (i.e. shorter than H). In other words, the FFD_RWA algorithm first routes 'longer' lightpaths which are harder to route, and then fills up the remaining space in each bin with 'shorter' lightpaths. This may lead to fewer wavelengths used.

4.2.4 *BFD_RWA*

The Best Fit Decreasing Routing and Wavelength Assignment algorithm sorts the lightpath requests in nonincreasing order of the lengths of their shortest paths SP_j in G, and then proceeds as BF_RWA.

Pseudocodes of the FF_RWA, BF_RWA, FFD_RWA and BFD_RWA algorithms are shown in Fig. 1. Some 'first fit' and 'longest path first' approaches have been used by wavelength assignment algorithms [4], but to the best of our knowledge have not been used to solve the routing subproblem, or for simultaneous routing and wavelength assignment. All four algorithms efficiently solve the static RWA problem, while the FF_RWA and BF_RWA algorithms can also be used for dynamic RWA. This makes sense since they are analogous to the 'on-line' FF and BF bin packing algorithms. Namely, in the *dynamic* RWA problem, lightpath requests in τ are not known *a priori*, but arrive unexpectedly. This means that lightpaths in τ are established in a specific order, i.e. in the order in which they arrive. If such is the case, the FF_RWA and BF_RWA algorithms simply establish lightpaths in the specified order according to their corresponding 'first fit' or 'best fit' strategies.

5 Lower bounds

Since the algorithms considered in this paper are heuristics which obtain upper bounds on the minimal objective function values, it is useful to have good lower bounds in order to assess the quality of the sub-optimal solutions. We use a lower bound for the number of wavelengths required which is similar to a lower bound developed for the virtual topology design problem presented in [16]. Stronger lower bounds may be found using more sophisticated methods, such as that in [17], but we use the lower bound presented here for its simplicity. Namely, the algorithms were tested for fairly large test problems, so using complex algorithms for finding better lower bounds was not practical. Furthermore, computational results demonstrate the efficiency of the suggested lower bound, particularly in denser networks. This bound on the number of wavelengths needed to establish a given set τ of n lightpath requests in a network with $|V|$ nodes and $|E_p|$ edges is

$$LB_W = max\{ \underset{i \in V}{max} \lceil \frac{\Delta_l(i)}{\Delta_p(i)} \rceil, \lceil \frac{\sum_{j=1}^{n} l(SP_j)}{2 * |E_p|} \rceil \}. \tag{1}$$

$\Delta_l(i)$ represents the logical degree of node i, i.e. the number of lightpaths for which node i is the source node. $\Delta_p(i)$ represents the physical degree of node i. $l(SP_j)$ is the length of the shortest path in G of lightpath request (s_j, d_j). The first element in (1) represents the maximum ratio of logical to physical degree of any node in G, rounded up to the first higher integer. If some node i has $\Delta_p(i)$ adjacent physical links and is the source node for $\Delta_l(i)$ lightpaths, at least one physical link will have $\lceil \frac{\Delta_l(i)}{\Delta_p(i)} \rceil$ lightpaths routed over it. Since lightpaths routed on the same physical links cannot be assigned the same wavelength, at least $\lceil \frac{\Delta_l(i)}{\Delta_p(i)} \rceil$ wavelengths are needed to route the corresponding lightpaths. The highest such ratio among all the nodes in the network is a lower bound on the number of wavelengths needed to perform RWA for set τ. In some cases, the second element in (1) may give a better lower bound. This element represents the minimum total physical hop length of all the lightpaths divided by the total number of links in the network. The minimum total physical hop length of the established lightpaths is the sum of the lengths of the shortest paths (in terms of hops) of all the lightpath requests. Since each edge in E_p represents 2 links, one in each direction, the total physical hop length is divided by $2 * |E_p|$.

A simple lower bound on the average physical length of the established lightpaths is equal to the average length of the shortest paths in G of all the lightpath requests in τ. We consider the lengths of lightpaths in terms of hops and refer to this lower bound as the Physical Hops lower bound, LB_{PH}. The bound is as follows.

$$LB_{PH} = \frac{\sum_{j=1}^{n} l(SP_j)}{n}. \tag{2}$$

6 Numerical Results

In order to determine the performance measures of the proposed algorithms, the *Greedy_EDP_RWA* [13] and the BF_RWA, FFD_RWA and BFD_RWA algorithms were implemented in C++ and run on a PC powered by a P4 2.8GHz processor. The FF_RWA algorithm was not implemented since it yields solutions equivalent to those obtained by the *Greedy_EDP_RWA*

Table 1: 100-node test networks with an average degree of 3: Lower bound and the average (*lowest, highest*) number of wavelengths used in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Netw.	P_l	Lightpath requests	LB_W	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2054	29	48.8 (*47,52*)	45.2 (*45,46*)	**45** (*45,45*)	**45** (*45,45*)
2		2034	24	49.6 (*49,51*)	49.1 (*49,50*)	**49** (*49,49*)	**49** (*49,49*)
3		2006	24	34.4 (*33,35*)	31.4 (*31,32*)	**29.9** (*29,30*)	30.5 (*30,31*)
4		2044	22	31.4 (*30,33*)	27.4 (*27,29*)	28 (*28,28*)	**26** (*25,27*)
5		2109	25	35.7 (*35,38*)	30.6 (*30,32*)	31.1 (*31,32*)	**29** (*28,30*)
1	0.4	4063	50	91.3 (*89,95*)	87.1 (*87,88*)	**87** (*87,87*)	**87** (*87,87*)
2		4038	41	96.9 (*96,98*)	96.1 (*96,97*)	**96** (*96,96*)	**96** (*96,96*)
3		3982	45	68.1 (*67,69*)	62.7 (*62,65*)	**60.6** (*60,61*)	61.2 (*61,62*)
4		4045	44	60 (*58,64*)	51 (*50,54*)	52.3 (*52,53*)	**48.7** (*48,49*)
5		4122	51	68.6 (*67,71*)	57.6 (*56,59*)	58.3 (*58,59*)	**56** (*54,58*)
1	0.6	6054	71	130 (*128,134*)	122.6 (*122,124*)	**122** (*122,122*)	**122** (*122,122*)
2		6020	65	127.9 (*127,129*)	**126** (*126,126*)	**126** (*126,126*)	**126** (*126,126*)
3		5999	65	100.3 (*98,104*)	92.5 (*92,95*)	**91** (*91,91*)	91.8 (*91,92*)
4		6048	62	86.1 (*85,88*)	73 (*72,74*)	77. 2 (*77,78*)	**71.4** (*71,72*)
5		6095	71	98 (*96,99*)	**82.3** (*79,86*)	84.4 (*84,86*)	85.7 (*83,88*)
1	0.8	8014	86	167.4 (*165,170*)	**161** (*161,161*)	**161** (*161,161*)	**161** (*161,161*)
2		8006	84	159.7 (*159,161*)	**158** (*158,158*)	**158** (*158,158*)	**158** (*158,158*)
3		7985	84	133.4 (*130,136*)	121.5 (*121,123*)	**120** (*120,120*)	**120** (*120,120*)
4		8008	86	115.5 (*114,119*)	97 (*95,99*)	100.8 (*100,101*)	**94.5** (*94,95*)
5		8046	88	127.2 (*123,131*)	**103.9** (*101,107*)	109 (*109,109*)	106.7 (*104,109*)
1	1.0	9900	99	207.5 (*205,210*)	**196** (*196, 196*)	**196** (*196,196*)	**196** (*196,196*)
2		9900	99	198.4 (*196,203*)	**196** (*196,196*)	**196** (*196,196*)	**196** (*196,196*)
3		9900	99	166.1 (*164,170*)	150.4 (*149,152*)	147.1 (*146,149*)	**146.1** (*146,147*)
4		9900	99	140.9 (*138,143*)	117.1 (*115,120*)	123.5 (*123,124*)	**114.6** (*114,115*)
5		9900	99	154.5 (*151,158*)	128.2 (*125,131*)	132.7 (*132,133*)	**129.6** (*127,132*)

algorithm. A series of random 100-node networks with average degrees of 3, 4, and 5 were created (5 networks per average degree). Random sets of lightpath requests were created for each test network with the probability P_l of there being a lightpath request between two nodes. The value of P_l ranged from 0.2 to 1.0, in 0.2 increments, for up to 9900 lightpath requests. The upper bound on the physical hop length of the established lightpaths, H, is set here to $\max(diam(G), \sqrt{|E_p|})$ as suggested in [13].

All four algorithms were run with 10 different seeds (i.e. 10 different permutations of τ) for each test case. The average, lowest and highest number of wavelengths of the solutions obtained by each algorithm were recorded. The average physical hop lengths of the established lightpaths were also found. The average number of wavelengths needed to successfully perform Routing and Wavelength Assignment by each of the algorithms for the test networks with an average degree of 3 are shown in Table 1. The lowest and highest solution values found are shown in parenthesis. The lower bound, LB_W, is also shown. The corresponding average lightpath lengths and the lower bound PH_{LB} are shown in Table 2. For test networks

Table 2: 100-node test networks with an average degree of 3: Lower bound and the average lightpath length in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Network	P_l	Lightpath requests	LB_{PH}	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2054	3.48	4.55	3.51	4.58	**3.48***
2		2034	3.40	4.44	3.43	4.46	**3.40***
3		2006	3.43	4.48	3.49	4.50	**3.47**
4		2044	3.39	4.50	3.65	4.52	**3.58**
5		2109	3.48	4.60	3.67	4.64	**3.65**
1	0.4	4063	3.48	4.54	3.49	4.56	**3.48***
2		4038	3.38	4.39	3.39	4.39	**3.38***
3		3982	3.44	4.46	3.48	4.49	**3.47**
4		4045	3.40	4.51	3.58	4.51	**3.53**
5		4122	3.51	4.62	3.63	4.64	**3.62**
1	0.6	6054	3.48	4.51	3.49	4.55	**3.48***
2		6020	3.38	4.37	3.39	4.40	**3.38***
3		5999	3.43	4.45	3.47	4.48	**3.46**
4		6048	3.42	4.50	3.57	4.52	**3.54**
5		6095	3.51	4.58	**3.59**	4.62	**3.59**
1	0.8	8014	3.48	4.53	3.49	4.54	**3.48***
2		8006	3.38	4.37	3.39	4.40	**3.38***
3		7985	3.45	4.45	**3.48**	4.49	**3.48**
4		8008	3.43	4.51	3.57	4.53	**3.54**
5		8046	3.51	4.59	**3.60**	4.61	**3.60**
1	1	9900	3.48	4.52	**3.49**	4.55	**3.49**
2		9900	3.39	4.37	3.40	4.41	**3.39***
3		9900	3.45	4.44	**3.48**	4.48	**3.48**
4		9900	3.43	4.49	3.56	4.53	**3.55**
5		9900	3.51	4.59	**3.61**	4.60	**3.61**

with an average degree of 4, the wavelengths and average lightpath lengths of the obtained solutions are shown in Tables 3 and 4, respectively. Tables 5 and 6 show the results obtained for test networks with an average degree of 5. The best obtained solution for each test case is marked in bold. If the obtained solution is equal to the lower bound, i.e. the obtained solution is surely optimal, it is marked as '*'.

The BF_RWA, FFD_RWA and BFD_RWA algorithms all perform significantly better than the *Greedy_EDP_RWA* algorithm for all cases with respect to the number of wavelengths used. The FFD_RWA and BFD_RWA algorithms perform best for this optimization criterion. In fact, the worst solution obtained by the FFD_RWA and BFD_RWA algorithms is better or equal to the best solution obtained by the *Greedy_EDP_RWA* algorithm in all cases. The worst solution obtained by the BF_RWA algorithm is better or equal to the best solution obtained by the *Greedy_EDP_RWA* algorithm in all but 2 cases for networks with an average

Table 3: 100-node test networks with an average degree of 4: Lower bound and the average (*lowest, highest*) number of wavelengths used in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Netw.	P_l	Lightpath requests	LB_W	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2116	21	24.3 (*23,25*)	21.7 (*21,24*)	**21*** (*21,21*)	**21*** (*21,21*)
2		2081	23	28.1 (*26,30*)	25.4 (*25,27*)	**25** (*25,25*)	**25** (*25,25*)
3		2067	21	25.8 (*24,27*)	24.4 (*24,26*)	**24** (*24,24*)	**24** (*24,24*)
4		2054	29	29.9 (*29,31*)	29.4 (*29,31*)	**29*** (*29,29*)	**29*** (*29,29*)
5		2125	32	33.8 (*32,36*)	**32*** (*32,32*)	**32*** (*32,32*)	**32*** (*32,32*)
1	0.4	4063	39	44.3 (*43,46*)	39.9 (*39,43*)	**39*** (*39,39*)	**39*** (*39,39*)
2		4047	46	50.9 (*49,53*)	47.2 (*47,48*)	**47** (*47,47*)	**47** (*47,47*)
3		4064	44	53.4 (*52,55*)	50.1 (*50,51*)	**50** (*50,50*)	**50** (*50,50*)
4		4063	47	51.4 (*50,52*)	48.3 (*47,50*)	**47*** (*47,47*)	**47*** (*47,47*)
5		4099	50	55.8 (*53,59*)	50.3 (*50,51*)	**50*** (*50,50*)	**50*** (*50,50*)
1	0.6	6017	60	66.9 (*63,69*)	61.1 (*61,62*)	**61** (*61,61*)	**61** (*61,61*)
2		5995	69	74.6 (*72,78*)	69.1 (*69,70*)	**69*** (*69,69*)	**69*** (*69,69*)
3		6054	65	75.4 (*73,78*)	67.8 (*67,71*)	**67** (*67,67*)	**67** (*67,67*)
4		6054	71	77.5 (*75,81*)	71.2 (*71,72*)	**71*** (*71,71*)	**71*** (*71,71*)
5		6113	66	77.9 (*76,83*)	67.6 (*66,70*)	**66*** (*66,66*)	**66*** (*66,66*)
1	0.8	7960	79	86.7 (*84,89*)	80.1 (*80,81*)	**80** (*80,80*)	**80** (*80,80*)
2		7988	80	89.8 (*88,93*)	81.8 (*81,83*)	**81** (*81,81*)	**81** (*81,81*)
3		8052	88	99.4 (*96,103*)	89.8 (*88,93*)	**88*** (*88,88*)	**88*** (*88,88*)
4		8014	86	94.4 (*91,99*)	86.6 (*86,89*)	**86*** (*86,86*)	**86*** (*86,86*)
5		8017	88	101.4 (*97,106*)	88.9 (*88,90*)	**88*** (*88,88*)	**88*** (*88,88*)
1	1.0	9900	99	108.3 (*106,110*)	99.9 (*99,102*)	**99*** (*99,99*)	**99*** (*99,99*)
2		9900	99	110.7 (*108,113*)	100.4 (*99,102*)	**99*** (*99,99*)	**99*** (*99,99*)
3		9900	99	120 (*118,123*)	105 (*103,109*)	**99*** (*99,99*)	**99*** (*99,99*)
4		9900	99	110.8 (*108,112*)	99.1 (*99,100*)	**99*** (*99,99*)	**99*** (*99,99*)
5		9900	99	122.7 (*119,125*)	106.7 (*105,109*)	**103.6** (*103,104*)	104.5 (*104,105*)

degree of 3, 4 cases for networks with an average degree of 4, and 7 cases for networks with an average degree of 5.

Since only those lightpaths whose shortest paths are equal in length are sorted randomly, the FFD_RWA and BFD_RWA algorithms usually perform the same for various permutations of τ. As a result, the worst solutions obtained by the FFD_RWA and BFD_RWA algorithms were in most cases their best solutions. These solutions were also better or equal to those obtained by the *Greedy_EDP_RWA* and BF_RWA algorithms. This seems to indicate that the FFD_RWA and BFD_RWA algorithms could be run only once and still obtain high quality solutions. The *Greedy_EDP_RWA* and BF_RWA algorithms, on the other hand, need to be run as multistart algorithms and even then they obtain inferior solutions.

As can be seen in Table 3, for networks with average degree 4, the *average* solutions obtained by the FFD_RWA and BFD_RWA algorithms were optimal in at least 16 out of the 25 test cases, in one case for the BF_RWA algorithm, and in zero cases for the *Greedy_EDP_RWA* algorithm. To obtain better results with the *Greedy_EDP_RWA* and BF_RWA algorithms,

Table 4: 100-node test networks with an average degree of 4: Lower bound and the average lightpath length in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Network	P_l	Lightpath requests	LB_{PH}	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2116	2.92	3.86	2.97	3.87	**2.93**
2		2081	2.95	3.88	2.98	3.87	**2.96**
3		2067	2.96	3.89	2.99	3.90	**2.96***
4		2054	3.05	4.03	3.11	4.03	**3.05***
5		2125	3.23	4.25	3.28	4.26	**3.24**
1	0.4	4063	2.91	3.84	2.93	3.85	**2.91***
2		4047	2.97	3.87	2.98	3.88	**2.97***
3		4064	2.97	3.87	2.98	3.88	**2.97***
4		4063	3.05	4.02	3.10	4.00	**3.05***
5		4099	3.24	4.23	3.29	4.25	**3.26**
1	0.6	6017	2.92	3.82	2.93	3.83	**2.92***
2		5995	2.96	3.85	2.97	3.87	**2.96***
3		6054	2.98	3.87	**2.98**	3.88	**2.98***
4		6054	3.05	4.00	3.07	4.00	**3.05***
5		6113	3.24	4.23	3.29	4.23	**3.27**
1	0.8	7960	2.93	3.83	2.94	3.84	**2.93***
2		7988	2.97	3.84	2.98	3.85	**2.97***
3		8052	2.98	3.86	2.99	3.87	**2.98***
4		8014	3.05	4.00	3.08	4.00	**3.05***
5		8017	3.24	4.22	3.27	4.23	**3.26**
1	1	9900	2.94	3.83	**2.94**	3.83	**2.94***
2		9900	2.97	3.85	**2.98**	3.85	**2.98**
3		9900	2.98	3.86	2.99	3.86	**2.98***
4		9900	3.05	4.00	3.07	4.00	**3.05***
5		9900	3.24	4.20	3.28	4.22	**3.27**

they could be run as multistart algorithms and then select the best found solution. The *best* solution obtained by the BF_RWA algorithm was optimal in 15 cases while the *best* solution obtained by the *Greedy_EDP_RWA* was optimal in only 2 cases. Table 5 indicates that for test networks with average degree 5, the *average* solutions obtained by the FFD_RWA and BFD_RWA algorithms were optimal in at least 14 out of the 25 test cases, while the BF_RWA and *Greedy_EDP_RWA* algorithms obtained optimal average solutions in two and zero cases, respectively. For these networks, the *best* solution obtained by the BF_RWA algorithm was optimal in 14 cases while the *best* solution obtained by the *Greedy_EDP_RWA* was optimal in only 3 cases. It is evident that sorting lightpath requests in nonincreasing order of their shortest paths helps obtain solutions using fewer wavelengths than establishing lightpaths at random.

The average length of the established lightpaths are compared in Tables 2, 4 and 6. Both the BF_RWA and BFD_RWA algorithms perform significantly better than the FF_RWA and

Table 5: 100-node test networks with an average degree of 5: Lower bound and the average (*lowest, highest*) number of wavelengths used in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Netw.	P_l	Lightpath requests	LB_W	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2116	20	21.7 (*21,23*)	20.3 (*20,22*)	**20*** (*20,20*)	**20*** (*20,20*)
2		2029	27	27.3 (*27,28*)	27.1 (*27,28*)	**27*** (*27,27*)	**27*** (*27,27*)
3		2081	22	24.8 (*24,27*)	**24** (*24,24*)	**24** (*24,24*)	**24** (*24,24*)
4		2067	19	20.1 (*19,22*)	19.1 (*19,20*)	**19*** (*19,19*)	**19*** (*19,19*)
5		2098	21	24.6 (*23,25*)	23.1 (*23,24*)	**23** (*23,23*)	**23** (*23,23*)
1	0.4	4063	37	39.4 (*38,41*)	37.2 (*37,38*)	**37*** (*37,37*)	**37*** (*37,37*)
2		3988	46	48.8 (*48,50*)	46.1 (*46,47*)	**46*** (*46,46*)	**46*** (*46,46*)
3		4047	46	47.2 (*46,48*)	46.1 (*46,47*)	**46*** (*46,46*)	**46*** (*46,46*)
4		4064	38	45.6 (*44,47*)	43.6 (*43,44*)	**43** (*43,43*)	**43** (*43,43*)
5		4100	38	47.7 (*47,49*)	**47** (*47,47*)	**47** (*47,47*)	**47** (*47,47*)
1	0.6	6017	60	61.8 (*61,64*)	**60*** (*60,60*)	**60*** (*60,60*)	**60*** (*60,60*)
2		5963	66	70.3 (*69,72*)	68.1 (*68,69*)	**68** (*68,68*)	**68** (*68,68*)
3		5995	69	71.1 (*70,73*)	69 (*69,69*)	**69*** (*69,69*)	**69*** (*69,69*)
4		6054	57	65.7 (*63,69*)	63.1 (*63,64*)	**63** (*63,63*)	**63** (*63,63*)
5		6088	60	64.1 (*62,66*)	62.2 (*62,63*)	**62** (*62,62*)	**62** (*62,62*)
1	0.8	7960	78	82.2 (*81,85*)	**79** (*79,79*)	**79** (*79,79*)	**79** (*79,79*)
2		7984	88	91.3 (*89,94*)	88.1 (*88,89*)	**88*** (*88,88*)	**88*** (*88,88*)
3		7988	79	83.5 (*81,85*)	80.1 (*80,81*)	**80** (*80,80*)	**80** (*80,80*)
4		8052	74	89.6 (*87,94*)	86.1 (*86,87*)	**86** (*86,86*)	**86** (*86,86*)
5		7994	78	83.7 (*82,87*)	81.1 (*81,82*)	**81** (*81,81*)	**81** (*81,81*)
1	1.0	9900	99	102.4 (*100,104*)	**99*** (*99,99*)	**99*** (*99,99*)	**99*** (*99,99*)
2		9900	99	109.9 (*107,113*)	100.1 (*99,102*)	**99*** (*99,99*)	**99*** (*99,99*)
3		9900	99	103.5 (*101,106*)	99.1 (*99,100*)	**99*** (*99,99*)	**99*** (*99,99*)
4		9900	99	105.2 (*103,107*)	99.1 (*99,100*)	**99*** (*99,99*)	**99*** (*99,99*)
5		9900	99	104.4 (*102,108*)	99.1 (*99,100*)	**99*** (*99,99*)	**99*** (*99,99*)

FFD_RWA algorithms, although the BFD_RWA algorithm performs best in all cases. In fact, the BFD_RWA algorithm obtains the optimal solution in at least 9, 17 and 25 cases for networks with average degrees 3, 4 and 5, respectively. Routing the lightpaths according to the 'best fit' strategy evidently leads to shorter lightpaths than using the 'first fit' strategy. For easier visualization of the obtained results, the average wavelengths and lightpath lengths of the solutions found for the test networks with an average degree of 4 are shown graphically in Fig. 2. Here the values for P_l ranged from 0.1 to 1.0 in 0.1 increments.

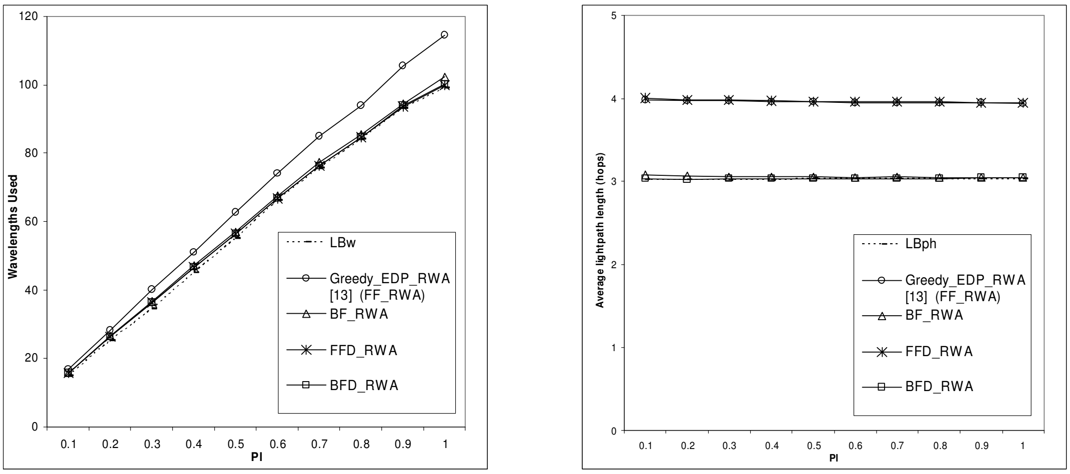
Furthermore, the algorithms were tested on a reference European core network topology shown in Fig. 3 which was designed as part of the COST Action 266 project [9]. P_l ranged from 0.1 to 1.0 in 0.1 increments. The results are shown in Fig. 4. The lower bound LB_W on the required number of wavelengths is not efficient for this network topology, so we assess the quality of the algorithms by comparing them to each other. Since the network is small and not many alternative paths are available, the algorithms performed fairly similar with respect to the number for wavelengths used (Fig. 4.(a)). However, the BF_RWA, FFD_RWA and

Table 6: 100-node test networks with an average degree of 5: Lower bound and the average lightpath length in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

Test Network	P_l	Lightpath requests	LB_{PH}	*Greedy_EDP _RWA [13] (FF_RWA)*	*BF_RWA*	*FFD_RWA*	*BFD_RWA*
1	0.2	2116	2.71	3.58	2.75	3.58	**2.71***
2		2029	2.83	3.67	2.84	3.68	**2.83***
3		2081	2.70	3.55	2.72	3.54	**2.70***
4		2067	2.67	3.50	2.71	3.51	**2.67***
5		2098	2.77	3.62	2.79	3.62	**2.77***
1	0.4	4063	2.70	3.55	2.73	3.55	**2.70***
2		3988	2.86	3.69	**2.86***	3.71	**2.86***
3		4047	2.71	3.55	2.73	3.54	**2.71***
4		4064	2.67	3.48	2.68	3.47	**2.67***
5		4100	2.74	3.58	2.75	3.58	**2.74***
1	0.6	6017	2.71	3.55	2.72	3.54	**2.71***
2		5963	2.84	3.67	2.85	3.69	**2.84***
3		5995	2.70	3.52	2.71	3.52	**2.70***
4		6054	2.68	3.48	2.70	3.48	**2.68***
5		6088	2.74	3.56	2.75	3.56	**2.74***
1	0.8	7960	2.71	3.55	2.73	3.54	**2.71***
2		7984	2.84	3.67	2.85	3.69	**2.84***
3		7988	2.71	3.53	2.72	3.52	**2.71***
4		8052	2.68	3.48	2.69	3.47	**2.68***
5		7994	2.74	3.56	**2.74***	3.55	**2.74***
1	1	9900	2.72	3.55	2.73	3.54	**2.72***
2		9900	2.83	3.65	**2.83***	3.67	**2.83***
3		9900	2.71	3.52	2.72	3.52	**2.71***
4		9900	2.69	3.47	2.70	3.47	**2.69***
5		9900	2.74	3.55	**2.74***	3.54	**2.74***

BFD_RWA algorithms performed the same or better than the *Greedy_EDP_RWA* algorithm in all cases. Fig. 4.(b) indicates that the 'best fit' algorithms again perform significantly better than the 'first fit' algorithms with respect to the average hop length. The BFD_RWA algorithm established the shortest lightpaths in all cases.

All four algorithms are very fast and highly tractable. The FFD_RWA and BFD_RWA algorithms are slower than the *Greedy_EDP_RWA*(FF_RWA) and BF_RWA algorithms by the time it takes to sort the lightpath demands. On the other hand, these algorithms are more robust and often give their best solutions in every run. As a result, these algorithms only need to be run once. The best and worst solution values obtained by the *Greedy_EDP_RWA* algorithm, on the other hand, vary significantly so this algorithm needs to be run as a multistart algorithm in order to obtain good results. This, of course, leads to much larger execution times. It should also be noted that the 'best fit' algorithms are somewhat slower



(a) (b)

Figure 2: 100-node test networks with an average degree of 4: Comparison of the (a) average number of wavelengths used and the (b) average lightpath length in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

with respect to the 'first fit' algorithms since they search among all the existing bins to find the 'best fit', while the first fit algorithms establishes the first found satisfactory route. When run on a PC powered by a P4 2.8GHz processor, the maximum execution time for the FFD_RWA and BFD_RWA algorithms for the 100 node networks with 9900 lightpath requests was less than 8 minutes. The maximum execution time for the FF_RWA and BF_RWA algorithms was under 6 minutes. For the European core network, all algorithms performed under half a second.

The following conclusions can be drawn from the obtained results. Sorting lightpaths in nonincreasing order of their shortest paths helps to obtain solutions using significantly fewer wavelengths. We can see from Tables 1, 3 and 5 that the advantage of sorting lightpaths becomes increasingly evident as the number of lightpath requests increases ($P_l \nearrow$). These are the cases where RWA is more challenging since we wish to establish a larger number of lightpaths. Routing lightpaths according to the 'best fit' strategy helps to consistently reduce lightpath hop length. The BFD_RWA algorithm, which both sorts lightpaths and uses the 'best fit' strategy, clearly performs best for all test cases.

Furthermore, recall that the *Greedy_EDP_RWA* and BF_RWA algorithms can be run for the dynamic Routing and Wavelength Assignment problem. For this problem, the mentioned algorithms are not run in multistart mode, but run once for each permutation of τ. As a result, we compare the *average* solution values obtained for the various permutations of τ. The BF_RWA algorithm performed significantly and consistently better than the *Greedy_EDP_RWA* algorithm with respect to both wavelengths and lightpath lengths. Using

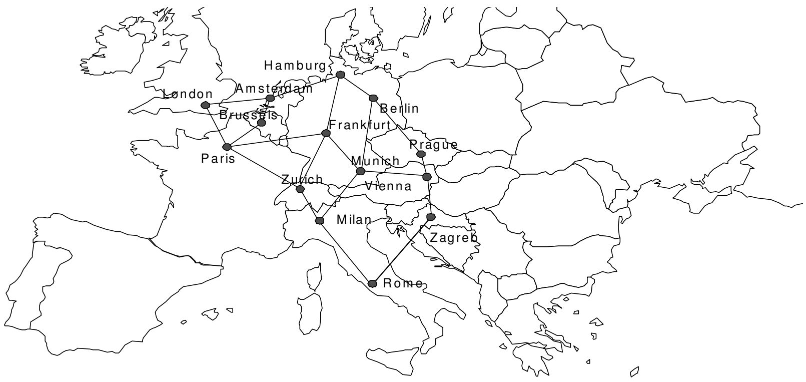
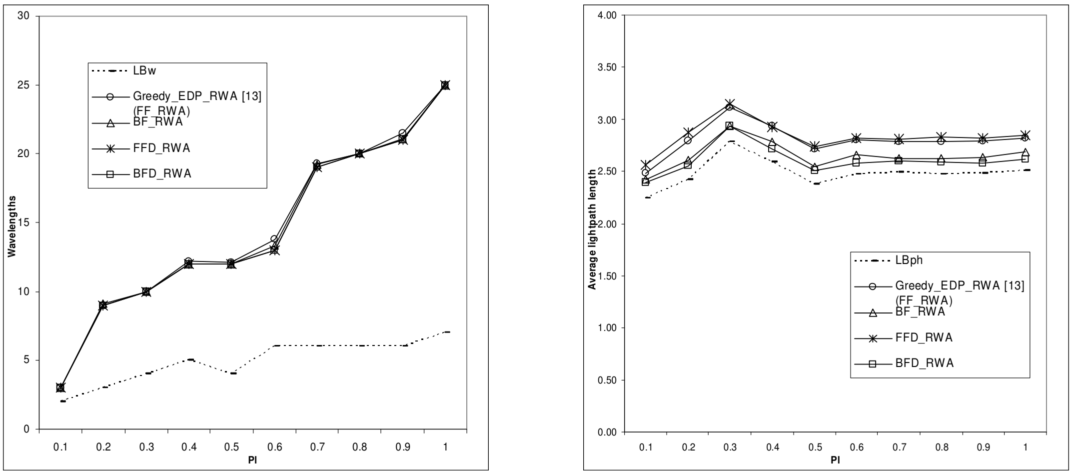


Figure 3: The hypothetical European core network [9].



(a)

(b)

Figure 4: The hypothetical European core network [9]: Comparison of the (a) average number of wavelengths used and the (b) average lightpath length in the solutions obtained by the *Greedy_EDP_RWA* algorithm (from [13]), and the BF_RWA, FFD_RWA and BFD_RWA algorithms proposed in this paper.

less wavelengths leaves more room for future lightpath requests. This decreases the chances that a lightpath request will be blocked due to the lack of available resources, which is a common objective criterion used to solve the dynamic RWA problem.

7 Conclusion

Successful solvability of the Routing and Wavelength Assignment (RWA) problem is mandatory for making efficient use of resources in wavelength routed optical networks. In this work, the bin packing problem is applied to optical networks to help develop highly efficient heuristic algorithms for the RWA problem. Suggested are methods of sorting and routing lightpaths which not only reduce the required number of wavelengths, but reduce the average physical length of established lightpaths as well. Numerical results indicate that the proposed methods obtain optimal or near optimal solutions in many cases, and significantly outperform an efficient existing algorithm from [13] for the same problem. Furthermore, the heuristics are robust and highly tractable and can thus be used to solve large problem instances in reasonable time. Further avenues of research will include developing similar algorithms for routing and wavelength assignment in networks with full or limited wavelength conversion. Networks equipped with a limited number of tranceivers and/or a limited number of wavelengths will also be considered.

References

[1] A. C. F. Alvim, F. Glover, C. C. Ribeiro and D. J. Aloise, "A hybrid improvement heuristic for the one-dimensional bin packing problem," *Journal of Heuristics,* 10 (2004), pp. 205-229.

[2] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignmnet in Large Wavelength-Routed Optical Newtorks," *IEEE Journal on Selected Areas in Communications* 14 (1996) pp. 903-908.

[3] I.Chlamtac, A. Ganz and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," *, IEEE Transactions on Communications* 40 (1992) pp.1171-1182.

[4] J. S. Choi, N. Golmie, F. Lapeyere, F. Mouveaux and D. Su, "A functional Classification Of Routing and Wavelength Assignment Schemes in DWDM networks: Static Case," *Proc. VII Int. Conf. on Optical Communication and Networks*, Jan. 2000.

[5] E. G. Coffman, Jr., J. Csirik, and G. Woeginger, "Bin Packing Theory," in: P. Pardalos and M. Resende (Eds.), Handbook of Applied Optimization, Oxford University Press, New York, 2002.

[6] E. G. Coffman, Jr., M. R. Garey and D. S. Johnson, "Bin Packing Approximation Algorithms: A Survey," in: D. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing Co., Boston, MA, 1996.

[7] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco 1979.

[8] E. Hyytiä and J. Virtamo, "Wavelength assignment and routing in WDM networks," *Nordic Telegraffic Seminar 14*, pp. 31-40, 1998.

[9] R. Inkret, A. Kuchar, B. Mikac "Advanced Infrastructure for Photonic Networks: Extended Final Report of COST Acteion 266", Zagreb: Faculty of Electrical Engineering and Computing, University of Zagreb, 2003, pp. 19-21.

[10] X. Jia, X.-D. Hu and D.-Z. Du, "Multiwavelength Optical Networks," Kluwer Academic Publishers, Norwell, MA, 2002.

[11] K. Lee, K. C. Kang, T. Lee and S. Park, "An Optimization Approach to Routing and Wavelength Assignment in WDM All-Optical Mesh Networks without Wavlength Conversion," *ETRI Journal* 24 (2) (2002) pp.131-141.

[12] G. Li and R. Simha, "The partition coloring problem and its application to wavelength routing and assignment," In *Proc. of Optical Networks Workshop*, Richardson, Texas, February 2000.

[13] P. Manohar, D. Manjunath and R. K. Shevgaonkar, "Routing and Wavelengths Assignment in Optical Networks From Edge Disjoint Path Algorithms," *IEEE Communication Letters* 6 (5) (2002) pp. 211-213.

[14] B. Mukherjee, "Optical Communication Networks." New York: McGraw-Hill, 1997.

[15] T. F. Noronha and C. C. Ribeiro, "Routing and wavelength assignment by partition coloring," *European Journal of Operational Research,* In Press, Corrected Proof, Available online 21 December 2004.

[16] R. Ramaswami and K. N. Sivarajan, "Design of Logical Topologies for Wavelength-Routed Optical Networks," *IEEE J. Select. Areas Commun.*, 14 (5) (1996) pp. 840 - 851.

[17] A. R. Sharafat, "The Most Congested Cutset: Deriving a Tight Lower bound for the Chromatic Number in the RWA problem," *IEEE Communication Letters*, 8 (7) (2004) pp. 473 - 475.