



**HAL**  
open science

# Routing and Wavelength Assignment in Optical Networks

Olivier Brun, Sami Baraketi

► **To cite this version:**

Olivier Brun, Sami Baraketi. Routing and Wavelength Assignment in Optical Networks. 2014. hal-01062321

**HAL Id: hal-01062321**

**<https://hal.science/hal-01062321>**

Submitted on 11 Sep 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# ROUTING AND WAVELENGTH ASSIGNMENT IN OPTICAL NETWORKS

O. BRUN, S. BARAKETI

## 1. INTRODUCTION

Transport technologies such as Synchronous Digital Hierarchy (SDH/SONET) and Asynchronous Transfer Mode (ATM) becomes increasingly speed-limited and can no longer respond to the demand for high bandwidth services (HDTV, video conferencing, electronic banking, multimedia applications, etc). They, through employing optical fiber, do not realize the full potential of the optical medium. The speed of these technologies are limited to few tens of Gbps due to the peak electronic speed of the network components, whereas a single mode fiber can carry data at very highest speeds. In order to increase the bandwidth of optical fiber, Wavelength division multiplexing (WDM) technology is evolved. It is a promising technology to effectively utilize the enormous bandwidth of optical fiber.

In wavelength division multiplexing technology the transmission spectrum of a fiber link can be divided into many protocol transparent channels. Multiple channels can be operated in a single fiber simultaneously at different wavelengths, providing for each channel the bandwidth that is compatible with current electronic processing speeds. These channels can be independently modulated to accommodate dissimilar data formats at various bit rates if necessary. By utilizing WDM in optical networks, we can achieve link capacities on the order of Tbps.

WDM networks are rapidly evolved as a powerful class of networks for use in wide area networks. These networks consist of optical switches that route a signal based on the identity of the input port (i.e. related overlying service) and the wavelength of the incoming signal. A WDM network is called also wavelength routed network [29] [30] since it employs wavelength routing technique. Access switches and end switches provide the electronic-to-optical conversion and vice versa to interface the optical network with electronic stations. Wavelength routing provides the network with the ability to identify and localize the traffic flow, thereby allowing the same wavelength to be reused in spatially disjoint segments of the network. In order to carry data from one access node to another, a connection needs to be set up at the optical layer similar to the case in a circuit-switched networks. This operation is performed by determining a path in the network connecting the source node to the destination node and by allocating a single free wavelength on all of the fiber links in the path. Such an all-optical path is referred as lightpath [7] [30], each lightpath can carry data at peak electronic speed. However, practical limitations on the

transmission technology and optical devices restrict the number of available wavelengths per fiber link, it is unlikely that a lightpath can be established between every pair of access nodes. The intermediate nodes in the path route the lightpath in the optical domain using wavelength-sensitive switches. A fundamental constraint in a wavelength-routed optical network is that two or more lightpaths traversing the same fiber link must be on different wavelengths so that they do not interfere with one another. A wavelength-routed network, which carries data from one access station to another without any intermediate optical-to-electrical conversion is referred as an all-optical wavelength-routed network. All-optical wavelength-routed networks will be the subject of our work. These networks have several benefits like the potential to accommodate the rapidly increasing bandwidth, improved network reliability, simpler network management, and are independent from modulation format and bit rate [11] [30].

Since the lightpaths are the basic switched entities of a wavelength-routed WDM network, their effective establishment and usage are crucial. Thus, it is important to propose efficient algorithms to select the routes for the requested connections and to assign wavelengths on each of the links along these routes. This is known as the routing and wavelength assignment problem. The routing and wavelength assignment problem (RWA) in optical networks considers a network where requests (i.e. lightpaths) can be transported on different optical wavelengths through the network. Each accepted request is allocated a path from its source to its sink, as well as a specific wavelength. Lightpaths routed over the same link must be allocated to different wavelengths, while lightpaths whose paths are link disjoint may use the same wavelength. Lightpaths that cannot be established due to constraint on wavelengths availability are said to be blocked.

## 2. LITERATURE SURVEY

Several works have studied the RWA problem in all-optical WDM networks and various contributions have been made through interesting algorithms. The problem consider a directed network  $G = (V, E)$  where  $V$  is the set of nodes representing the switches of the physical network and  $E$  is the set of edges representing the fiber links of the physical network. Given a set of requests for all-optical connections or lightpaths between node-pairs and a set of available wavelengths, the problem is to find routes from the source nodes to their respective destination nodes and assign wavelengths to these routes.

The RWA problem in WDM networks can be categorized into two types based on traffic arrivals.

(1) Static Lightpath Establishment (SLE) : The traffic is static and the set of connection requests is known in advance. This kind of problem pertains to the planning phase of the WDM network. The algorithms proposed for solving the static RWA problem are referred to as Offline algorithms. Static RWA is known to be an NP-hard optimization problem [7] since it is considered as a special case of the integer multicommodity flow (MCF) problem [21] [13] with additional specific constraints and can be formulated as an integer linear program (ILP).

(2) Dynamic Lightpath Establishment (DLE) in the case of dynamic traffic : The traffic is dynamic and the connection requests arrive sequentially, one by one, at random times over an infinite time horizon. The DLE problem is posed in the operational phase where each network resource must be managed efficiently. The algorithms proposed for solving the dynamic RWA problem are referred to as Online algorithms. A review on dynamic RWA algorithms can be found in [40].

In our study we will focus on the static RWA problem and the offline algorithms that proposed in literature to solve this kind of problem. Offline RWA is more difficult than online RWA since it aims at jointly optimizing the lightpaths used by the connections in the same way that the multicommodity flow problem is more difficult than the shortest-path problem in general networks. We classify the previous works in three classes. For each class we discuss the already proposed algorithms and the adopted approaches.

**2.1. Joint RWA ILP-based algorithms.** Many ILP-based formulations have been proposed in literature for routing and wavelength assignment problems jointly. [15] provides different ILP formulations (path-based, edge-based and arc-based formulations). The authors proposed a synthesis of the mathematical models for symmetrical systems where bandwidth requirements are similar on the downstream and upstream directions. Several objectives have been considered: (a) minimizing the blocking rate or maximizing the number of accepted requests given a fixed number of available wavelengths, (b) minimizing the number of used wavelengths assuming that all connection requests can be accepted, (c) minimizing the congestion that is expressed through the minimization of the maximum number of wavelengths on a given fiber link, (d) minimizing the network load defined as the ratio of used wavelengths over the overall potential number of wavelengths. A continuous relaxations have been also proposed. The authors of [5] developed an ILP model for RWA problem using other optimization criteria like minimizing the number of wavelength conversions or minimizing the hop count. [34] proposed two direct RWA ILP formulations : basic formulation in which the aim is to minimize the maximum number of wavelengths per fiber link, and extended formulataion that try to minimize the overall number of used wavelengths. A source aggregation is considered to reduce the number of constraints and improve the two mathematical models. In [14], the authors proposed a new formulation that addresses the complete trafç grooming problem, including topology design as well as routing and wavelength assignment (RWA) of lightpaths. In Other works, the authors used some efficient techniques in order to scale their RWA ILP formulations to problem instances encountered in practice. A link selection techniques were considered in [24] to reduce the size of the link-based formulation in terms of both the number of variables and the number of constraints. Column Generation technique was also used in [16] [20] to improve the RWA formulation. In [23], the authors demonstrated that their new path-based formulation achieves a decrease of up to two orders of magnitude in running time compared to existing formulations. Since the majority of proposed ILP were very hard to solve, a corresponding relaxed linear programs have been used to get bounds on the optimal value that can be achieved. As an example, The LP-relaxation formulation proposed in [27], and

also considered in [9], is able to produce exact RWA solutions in many cases, despite the absence of integrality constraints. In [10], the authors presented an algorithm for solving the static RWA problem based on a LP-relaxation formulation that provides integer optimal solutions despite the absence of integrality constraints for a large subset of RWA input instances. The RWA formulation was then extended in order to take into consideration the physical layer impairments and account for the interference among lightpaths. Iterative fixing and Rounding techniques have been also used to provide an integer solution for the relaxed problem.

**2.2. RWA decomposition-based algorithms.** Another known approach in literature is to break the RWA problem in the two constituent subproblems and solves them individually and sequentially. This approach consists on two steps, by first finding routes for all requested connections and secondly searching for appropriate wavelength assignment [39] [37]. Note that both subproblems are NP-hard: The routing subproblem for a set of connections corresponds to a multicommodity flow problem, while wavelength assignment corresponds to a graph coloring problem. Several works have been adopted a decomposition-based algorithms. In [34] [2] [9], the authors choose a decomposition technique which handles the first step with a ILP program which assigns paths to the demands while minimizing the maximal number of demands routed over a link. The second step is expressed as a graph coloring problem where the nodes are demands and disequality constraints (links) are imposed between any two demands which are routed over the same link. The final solution is an approximate solution of the original complete RWA problem that can be not optimal. Works [2] [9], formulated the routing sub-problem as a continuous multicommodity flow problem and applied a randomized rounding technique to provide an integer solution in which the objective function takes on a value close to the optimum of the rational relaxation. A performance comparison between a RWA ILP-based algorithm and a decomposition-based algorithm was made in [9]. However, [39] synthesized a lot of known approaches for resolving the routing sub-problem. Fixed routing, fixed-alternate routing and adaptive routing were described. Ten dynamic wavelength assignment heuristics were also discussed. the authors said that these heuristics may also be applied to the static wavelength assignment problem by ordering the lightpaths and then sequentially assigning wavelengths to the ordered lightpaths. Note that all the proposed decomposition techniques, previously cited, suffer from two major drawbacks: (1) The approximate solution, obtained as a result of the problem decomposition, is often not optimal (2) The optimal solution of the joint RWA might not be included in the solutions provided by the algorithms used for the two subproblems.

**2.3. RWA heuristics.** For static lightpath establishment (SLE), several heuristic algorithms have been proposed for establishing a maximum number of lightpaths from a given set of request [12] [26] [42]. However, most of these old algorithms are based on traditional circuit-switched networks where routing and wavelength assignment steps are decoupled. Several recent studies have been focused on solving the joint RWA problem by sophisticated heuristics. In [35, 33, 6, 36], a lot of proposed heuristics were presented and evaluated. In

the following, we summarize and describe the most known heuristics used for solving the joint RWA problem.

- Shortest First Fixed Path (SFFP) [33]: M. shiva Kumar used wavelength-graph or layered-graph to find routes and wavelengths for the given lightpath set. In the layered graph model the physical network  $G = (V, E)$  is replicated  $|\mathcal{W}|$  times, each sub graph  $G = (V_\lambda, E_\lambda)$ ,  $\lambda \in \mathcal{W}$  corresponding to the given physical network on a particular wavelength. The objective here was to maximize the network throughput. The heuristic algorithm finds the shortest path for all the given node-pairs by using Dijkstra's algorithm in the given physical network topology. Then the requested connections are arranged in non-decreasing order of their path lengths. Now heuristic algorithm routes lightpaths sequentially on the wavelength-graph in a single layer with the shortest path. If a route is found with finite cost, then the lightpath is established, and the wavelength-graph is updated by assigning infinite cost to the edges along which the request is routed. If a route with minimum cost is not found for a node pair then the request is skipped, and the next request from the sorted list is considered. In the second phase the algorithm finds the shortest available path on the residual wavelength-graph for the lightpath requests that were skipped in the first phase. Thus by assigning a wavelength to the shortest path first maximizes the number of lightpaths established, which is equivalent to maximizing the network throughput.
- Longest First Fixed Path (LFFP) or Longest-path First [7] [35]: In this heuristic, wavelengths are assigned to the longest lightpath first. A comparison with SFFP was made in [33] based on the number of established lightpaths. The results showed that The number of lightpaths established by SFFP is more after a certain size of the lightpath request set. However, the average fiber utilization obtained by LFFP is more when compared to SFFP. This is because, by establishing long lightpaths first, would result better wavelength reuse.
- Minimum Number of Hops (MNH) [3] [35]: Baroni and Bayvel proposed an MNH algorithm for minimizing the maximum load per link in arbitrarily connected networks. In MNH, each node-pair of the given set of connection requests is firstly assigned one of its shortest paths. Then, alternate shortest paths are examined for a possible better path and the previously assigned path will be replaced by the alternate path if the load of the most congested link is reduced. This process is repeated for all the node-pairs and stops when no substitutions are possible. Results in [35] showed that the MNH provided a more efficient routing with a minimum number of used wavelengths than LFFP.
- Longest First Alternate Path (LFAP) [6] [35]: In LFAP, the RWA problem is formulated as a knapsack problem. Wavelengths are treated as knapsacks, each of which can hold more than one lightpath. Lightpaths are treated as items and more

than one lightpath can share the same wavelength on condition that no two lightpaths pass through the same link. The LFAP algorithm assigns a wavelength to a longer lightpath with higher priority and attempts to maximize the number of lightpaths per wavelength. More precisely, wavelengths are added one by one until all the lightpaths for the given set of requests are established. For each newly added wavelength, the longest lightpath among those of the given requests is established. Then, the shorter lightpaths will be checked one by one. If no lightpath can be established, alternate paths are searched. If no lightpath can be established any more, a new wavelength will be added and the searching process is repeated. After establishing each lightpath, the network topology is modified by removing the links used by the newly established lightpath. The results in [35] showed that the performance (i.e. number of wavelengths required) of LFAP is much better than LFFP and MNH. However, LFAP spends more time than LFFP in order to provide solution.

- Heaviest Path Load Deviation (HPLD) [35]: In HPLD, the RWA problem is formulated as a routing problem where the link cost is determined based on the load (utilization) of each link. The HPLD algorithm attempts to re-route some lightpaths that pass through the heaviest link in order to minimize the number of wavelengths. More precisely, HPLD algorithm search to deviate the load of the most loaded link to other less loaded path so that the maximum number of wavelengths used in the network is reduced. That is, the HPLD algorithm tries to re-route some lightpaths that pass through the heaviest link. The HPLD algorithm employs the shortest path routing technique to solve this problem based on the network graph. The weight function of a link (link cost) is determined by the link load. The results in [35] showed that HPLD is a bit more efficient than LFAP in term of the number of used wavelengths. However, it needs more time than LFAP to find a feasible solution.

Note that the four first following heuristics were developed by applying classical bin packing algorithms [19]. Lightpath requests represented items, while copies of graph  $G$  represented bins. Each copy of  $G$ , referred to as bin  $G_i$ ,  $i = 1, \dots, |W|$ , corresponds to one wavelength. The authors in [36] showed that their heuristics were tested on a series of large random networks and compared with an efficient previous algorithm for the same problem. Results indicated 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.

- First Fit (FF-RWA) [36], called also Greedy-EDP-RWA in [25] based on the bounded-length greedy algorithm for Edge-Dijoint Paths problem: 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 place (i.e. 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.

- Best Fit (BF-RWA) [36]: BF-RWA routed lightpaths in the bin into which they fit “best”. The algorithm considered 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  $k = 1, \dots, B$ , 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  $i = 1, \dots, B$ ), 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
- First Fit Decreasing (FFD-RWA) [36]: FFD-RWA algorithm sorts the lightpath requests in non-increasing 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  $G_1$ . 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.
- Best Fit Decreasing (BFD-RWA) [36]: BFD-RWA algorithm sorts the lightpath requests in non-increasing order of the lengths of their shortest paths  $SP_j$  in  $G$ , and then proceeds as BF-RWA. The obtained results in [36] showed that BFD-RWA provided more better solution (i.e. number of required wavelengths) than the three other algorithms.
- A weight based Edge Disjoint Path (WEDP) algorithm has been also proposed in [32] to solve the RWA problem. The authors showed that WEDP performed better than a greedy MEDP algorithm used for solving the same problem.



TABLE 1. Summary of static RWA algorithms

Problems	Approaches	Comments	References
Joint RWA	ILP Formulation (including relaxed formulation)	NP-complete	[15, 5, 34, 14, 24, 23] [27, 9, 10]
	SFFP LFFP MNH LFAP HPLD FF-RWA BF-RWA FFD-RWA BFD-RWA	Heuristics  Objective: Minimizing the number of required wavelengths	[7, 35, 33, 6, 3]  [36, 25]
Routing	ILP Formulation (including relaxed formulation)	NP-complete	[31, 34, 2, 9]
	Fixed routing Alternate routing		[39, 22, 28]
Wavelength Assignment	Graph coloring	NP-complete (routes are known)	[39, 34, 2, 9]
	Random LU (SPREAD) FF MP (multi-fiber) MU (PACK) LL (multi-fiber) M $\Sigma$ RCL	Heuristics used with Fixed Routing approach	[39, 4, 17, 18, 38, 41]

The Table 1 summarizes all the previous described works. Note that we illustrate only static/offline RWA cases.

### 3. MINIMIZATION OF THE NUMBER OF LIGHPATHS

Given a set of traffic demands, our goal is to minimize the number of new lighpaths to be created in the optical network for supporting these demands. We address this problem by considering only the edge nodes of the Optical Transport Network (OTN). We view a lighpath between nodes  $u$  and  $v$  as a directed edge between these nodes, with a certain capacity representing the maximum amount of traffic that a lighpath can accomodate. We assume that the number of lighpath between any two nodes is given by some large integer value  $K$ . We show that the problem of minimizing the number of lighpaths can then be formulated as that of routing demands in a multigraph with the goal of minimizing the number of used links. We propose a simple heuristic to solve this problem.

**3.1. Problem statement.** We consider a network represented by a multidigraph  $G := (V, E, s, t)$ , where

- $V$  is a set of vertices or nodes,
- $E$  is a set of edges or lines,
- $s : E \rightarrow V$  assigns to each edge its source node,
- $t : E \rightarrow V$  assigns to each edge its target node.

The network is such that there are  $K$  edges between any two pairs of nodes, that is the cardinality of the set

$$\{e \in E : s(e) = i, t(e) = j\},$$

is  $K$  for all  $i, j \neq i$  in  $V$ . An example of such a network is shown in Figure 1. It is assumed that the capacity of each edge  $e \in E$  is  $C$  units of bandwidth.

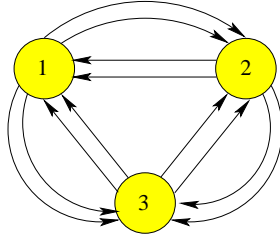


FIGURE 1. Simple example of network with 3 nodes and 2 directed edges between each pair of nodes.

We are given a set  $\mathcal{D}$  of traffic demands that have to be routed in the network. For each demand  $d \in \mathcal{D}$ , we let  $\lambda_d$  be the traffic volume of demand  $d$ ,  $s_d$  its source node and  $t_d$  its destination node. Traffic demand  $d$  ships its flow by splitting its demand  $\lambda_d$  over a set of paths  $\Pi(d)$ . We assume that the set  $\Pi(d)$  contains all simple path between  $s_d$  and  $t_d$ . Since the network is symmetric, the number of paths is the same for all demands and it will be denoted by  $S$  in the following. Let  $x_{d,\pi}$  denote the amount of traffic sent by demand  $d$  over path  $\pi$ . A routing strategy for demand  $d \in \mathcal{D}$  is a vector  $\mathbf{x}_d = (x_{d,\pi})_{\pi \in \Pi(d)}$  in  $\mathbb{R}_+^S$  such that  $\sum_{\pi \in \Pi(d)} x_{d,\pi} = \lambda_d$ . We let  $\mathcal{X}_d$  denote the set of all routing strategies for demand  $d$ :

$$\mathcal{X}_d = \left\{ \mathbf{x}_d \in \mathbb{R}_+^S : \sum_{\pi \in \Pi(d)} x_{d,\pi} = \lambda_d \right\}.$$

The vector  $\mathbf{x} = (\mathbf{x}_d)_{d \in \mathcal{D}}$  will be called a routing strategy. It belongs to the product strategy space  $\mathcal{X} = \bigotimes_{d \in \mathcal{D}} \mathcal{X}_d$ . For each edge  $e \in E$ , we define the function  $y_e : \mathcal{X} \rightarrow \mathbb{R}_+$  by

$$y_e(\mathbf{x}) = \sum_{d \in \mathcal{D}} \sum_{\pi \in \Pi(d)} \delta_e^\pi x_{d,\pi}.$$

where  $\delta_e^\pi = 1$  if  $e \in \pi$ , and 0 otherwise. In other words,  $y_e(\mathbf{x})$  represents the amount of traffic flowing through link  $e$  under strategy  $\mathbf{x}$ . A routing strategy is feasible if  $y_e(\mathbf{x}) \leq C$  for all links  $e$ . In the following, we say that link  $e$  is used under strategy  $\mathbf{x}$  if and only if  $y_e(\mathbf{x}) > 0$ .

We wish to find a feasible routing strategy which minimizes the number of used links. This amounts to solving the following optimization problem:

$$\begin{aligned}
 \text{(OPT)} \quad & \text{minimize } v(\mathbf{x}) = \sum_{e \in E} \mathbb{1}_{\{y_e(\mathbf{x}) > 0\}} \\
 & \text{subject to} \\
 & \quad \mathbf{x} \in \mathcal{X}, \\
 & \quad y_e(\mathbf{x}) \leq C, \quad \forall e \in E.
 \end{aligned}$$

It is immediate to see that problem (OPT) can be formulated as the following integer programming problem:

$$\begin{aligned}
 \text{(OPT-LP)} \quad & \text{minimize } \sum_{e \in E} b_e \\
 & \text{subject to} \\
 & \quad y_e \leq C \cdot b_e, \quad \forall e \in E, \\
 & \quad y_e = \sum_{d \in \mathcal{D}} \sum_{\pi \in \Pi(d)} \delta_e^\pi x_{d,\pi}, \quad \forall e \in E, \\
 & \quad \sum_{\pi \in \Pi(d)} x_{d,\pi} = \lambda_d, \quad \forall d \in \mathcal{D}, \\
 & \quad x_{d,\pi} \geq 0, \quad \forall \pi \in \Pi(d), \forall d \in \mathcal{D}, \\
 & \quad y_e \geq 0, \quad \forall e \in E, \\
 & \quad b_e \in \{0, 1\}, \quad \forall e \in E.
 \end{aligned}$$

In the following, we shall assume that  $K$  is sufficiently large for the above problem to have a feasible solution. More precisely, we shall assume that

$$K \geq \left\lceil \frac{\sum_d \lambda_d}{C} \right\rceil.$$

We note that problem (OPT-LP) involves binary variables and is thus difficult to solve. We propose in the following section a simple successive approximation heuristic to find an approximate solution.

**3.2. Successive approximation heuristic.** Before describing the successive approximation algorithm in Section 3.2.2, we first establish in Section 3.2.1 some results regarding the optimal routing strategy of a single demand when the routing strategies of the other demands are given.

3.2.1. *Optimal routing strategy for a single traffic demand.* Denote by  $\mathbf{x}_{-d}$  the vector  $(\mathbf{x}_f)_{f \neq d}$ , that is the routing strategy obtained by routing all traffic demands but demand  $d$ . It is assumed that this routing strategy is fixed and feasible. It can be viewed as a partial solution to problem (OPT). Define  $E^-$  as the set of links that are used under strategy  $\mathbf{x}_{-d}$ , that is

$$E^- = \{e \in E : y_e(\mathbf{x}_{-d}) > 0\},$$

and define  $E^+$  as  $E^+ = E \setminus E^-$ . We do not make explicit the dependance on  $d$  in order to simplify notations. Let  $c_e = C - y_e(\mathbf{x}_{-d})$  be the residual capacity for demand  $d$  on link  $e$ . The minimum-cost solution that can be obtained from the partial solution  $\mathbf{x}_{-d}$  is then obtained by solving the following optimization problem:

$$\begin{aligned} \text{(OPT-}d\text{)} \quad & \text{minimize } v_d(\mathbf{x}_d) = \sum_{e \in E^+} \mathbb{1}_{\{y_e(\mathbf{x}_d) > 0\}} \\ & \text{subject to} \\ & \mathbf{x}_d \in \mathcal{X}_d, \\ & y_e(\mathbf{x}_d) \leq c_e, \quad \forall e \in E. \end{aligned}$$

As before, we note that the above problem can be formulated as a mixed linear programming problem:

$$\begin{aligned} \text{(OPT-}d\text{-LP)} \quad & \text{minimize } \sum_{e \in E^+} \beta_e \\ & \text{subject to} \\ & z_e \leq C \cdot \beta_e, \quad \forall e \in E^+, \\ & z_e \leq c_e, \quad \forall e \in E^-, \\ & z_e = \sum_{\pi \in \Pi(d)} \delta_e^\pi x_{d,\pi}, \quad \forall e \in E, \\ & \sum_{\pi \in \Pi(d)} x_{d,\pi} = \lambda_d, \\ & x_{d,\pi} \geq 0, \quad \forall \pi \in \Pi(d), \\ & z_e \geq 0, \quad \forall e \in E, \\ & \beta_e \in \{0, 1\}, \quad \forall e \in E. \end{aligned}$$

In the following, we denote by  $\mathbf{x}_d^*$  an optimal solution of problem (OPT- $d$ ). Proposition 1 gives an explicit expression for the cost of such an optimal routing strategy.

**Proposition 1.** *The optimal value of problem (OPT- $d$ ) is*

$$v_d(\mathbf{x}_d^*) = \left\lceil \frac{\lambda_d - \lambda(E^-)}{C} \right\rceil,$$

where  $\lambda(E^-)$  is the optimal value of the following linear program:

$$\begin{aligned}
(\text{Max-Flow}) \quad & \text{maximize } \lambda \\
& \text{subject to} \\
& z_e \leq c_e, \quad \forall e \in E^-, \\
& z_e = \sum_{\pi \in \Pi(d)} \delta_e^\pi x_{d,\pi}, \quad \forall e \in E^-, \\
& \sum_{\pi \in \Pi(d), \pi \subset E^-} x_{d,\pi} = \lambda, \\
& \lambda \leq \lambda_d, \\
& x_{d,\pi} \geq 0, \quad \forall \pi \in \Pi(d), \pi \subset E^- \\
& z_e \geq 0, \quad \forall e \in E^-,
\end{aligned}$$

*Proof.* Let  $n = \left\lceil \frac{\lambda_d - \lambda(E^-)}{C} \right\rceil$ . The proof is in two parts. We first show that for each routing strategy  $\mathbf{x}_d \in \mathcal{X}_d$  such that  $y_e(\mathbf{x}_d) \leq c_e, \forall e$ , we have  $v_d(\mathbf{x}_d) \geq n$ . We then prove that there exists a routing strategy for demand  $d$  whose cost is  $n$ .

Consider a solution  $\mathbf{x}_d \in \mathcal{X}_d$  such that  $y_e(\mathbf{x}_d) \leq c_e, \forall e$ . Let  $\alpha = \sum_{\pi \in \Pi(d), \pi \subset E^-} x_{d,\pi}$  be the amount of traffic routed on paths in the subgraph  $(V, E^-)$ . By definition of  $\lambda(E^-)$ , we have  $\alpha \leq \lambda(E^-)$ . Since  $\frac{y_e(\mathbf{x}_d)}{C} \leq 1$  for all  $e \in E^+$ , we have

$$\begin{aligned}
v(\mathbf{x}_d) &= \sum_{e \in E^+} \mathbb{1}_{\{y_e(\mathbf{x}_d) > 0\}}, \\
&= \sum_{e \in E^+} \left\lceil \frac{y_e(\mathbf{x}_d)}{C} \right\rceil, \\
&\geq \left\lceil \frac{1}{C} \sum_{e \in E^+} y_e(\mathbf{x}_d) \right\rceil.
\end{aligned}$$

However

$$\begin{aligned}
 \sum_{e \in E^+} y_e(\mathbf{x}_d) &= \sum_{e \in E^+} \sum_{\pi \in \Pi(d)} \delta_e^\pi x_{d,\pi} \\
 &= \sum_{\pi \in \Pi(d)} \left( \sum_{e \in E^+} \delta_e^\pi \right) x_{d,\pi} \\
 &\geq \sum_{\pi \in \Pi(d), \pi \cap E^+ \neq \emptyset} x_{d,\pi} \\
 &\geq \lambda_d - \alpha,
 \end{aligned}$$

from which we obtain

$$v(\mathbf{x}_d) \geq \left\lceil \frac{\lambda_d - \alpha}{C} \right\rceil \geq \left\lceil \frac{\lambda_d - \lambda(E^-)}{C} \right\rceil = n.$$

We thus conclude that any feasible solution to (OPT- $d$ ) has a cost greater than or equal to  $n$ .

We now turn to the second part of the proof. Let  $\mathbf{x}_d^-$  be the optimal solution of problem (Max-Flow). If  $n = 0$ , i.e., if  $\lambda(E^-) = \lambda_d$ , then the routing strategy  $x_d^*$  defined by

$$x_{d,\pi}^* = \begin{cases} x_{d,\pi}^- & \text{if } \pi \subset E^-, \\ 0 & \text{otherwise.} \end{cases}$$

is clearly an optimal solution to (OPT- $d$ ). Otherwise, choose arbitrarily  $n$  edges  $e_1, \dots, e_n \in E^+$  such that  $s(e_k) = s_d$  and  $t(e_k) = t_d$  for  $k = 1, \dots, n$  and let  $\pi_k$  be the path  $\pi_k = \{e_k\}$  for  $k = 1, \dots, n$ . Consider the routing strategy  $x_d^* \in \mathcal{X}_d$  defined as follows:

$$x_{d,\pi}^* = \begin{cases} x_{d,\pi}^- & \text{if } \pi \subset E^-, \\ \frac{\lambda_d - \lambda(E^-)}{n} & \pi = \pi_1, \dots, \pi_n \\ 0 & \text{otherwise.} \end{cases}$$

We clearly have  $x_{d,\pi}^* \geq 0$  for all paths  $\pi \in \Pi(d)$ . Moreover,

$$\begin{aligned}
 \sum_{\pi \in \Pi(d)} x_{d,\pi}^* &= \sum_{\pi \in \Pi(d), \pi \subset E^-} x_{d,\pi}^* + \sum_{k=1}^n x_{d,\pi_k}^*, \\
 &= \lambda(E^-) + \lambda_d - \lambda(E^-), \\
 &= \lambda_d,
 \end{aligned}$$

which proves that  $\mathbf{x}^* \in \mathcal{X}_d$ . Note that by definition of  $\mathbf{x}_d^-$ , we have  $y_e(\mathbf{x}_d^*) = y_e(\mathbf{x}_d^-) \leq c_e$  for all  $e \in E^-$ . Moreover, since  $\frac{\lambda_d - \lambda(E^-)}{n} \leq C$ , we have  $y_{e_k}(\mathbf{x}_d^*) \leq C = c_e$  for  $k = 1, \dots, n$ . We thus conclude that  $y_e(\mathbf{x}_d^*) \leq c_e, \forall e \in E$ . Since the routing strategy  $\mathbf{x}_d^*$  uses  $n$  links in

$E^+$ , we conclude that  $v_d(\mathbf{x}_d^*) = n$ , and thus there exists at least one feasible solution to (OPT- $d$ ) whose cost is  $n$ .  $\square$

We note that the second part of the proof of Proposition 1 provides an algorithm to find an optimal solution to problem (OPT- $d$ ):

- Step 1.** Solve Problem (Max-Flow) in order to find the value of  $\lambda(E^-)$  and the amount of traffic to be routed on each path  $\pi$  between  $s_d$  and  $t_d$  in the subgraph  $(V, E^-)$ . We note that the structure of this problem is that of a standard maximum flow problem, for which very efficient algorithms are known [1].
- Step 2.** Choose arbitrarily  $n = \left\lceil \frac{\lambda_d - \lambda(E^-)}{C} \right\rceil$  edges in  $E^+$  between  $s_d$  and  $t_d$ , and split evenly the traffic  $\lambda_d - \lambda(E^-)$  among these links.

**3.2.2. Successive approximation algorithm.** The successive approximation algorithm is described in Figure 2. At each iteration, this algorithm routes optimally a single traffic demand assuming the routing strategies of the other demands are fixed. The algorithm stops when it is no more possible to decrease the number of used links by re-routing a single traffic demand.

**Lemma 1.** *The successive approximation algorithm converges in a finite number of steps.*

*Proof.* Define a *round* to be a sequence of iterations of the algorithm in which each traffic demand is rerouted exactly once (Steps 4-19 in Figure 2). Once an order is fixed in the first round, it is assumed to be the same in each subsequent round (the order in which the traffic demands are routed in the first-round can be arbitrary). Observe that if the algorithm does not stop at the end of a round, then the number of used links at the end of the round is lower by at least one than the number of used links at the end of the previous round. Hence, the number of used links at the end of each round is a strictly decreasing sequence. Since this number has to be positive, the algorithm converges in a finite number of rounds.  $\square$

**3.3. Experiments and results.** In this section, several experiments on the optimization problem for traffic routing were performed in order to evaluate and validate the effectiveness of the proposed solutions. The CPLEX solver, from IBM-ILOG society, is used for solving the integer linear program (OPT-LP). The aim of the experiments here is to analyze the performance and compare the proposed methods. The comparison is performed between the ILP-based algorithm (OPT-LP) and the successive approximation algorithm illustrated in Fig. 2.

In the following sections, the performance comparison between the algorithms is done based on three criteria. The first and second ones are related to the computation time and the memory consumption, respectively. The third one represents the routing strategy cost, which reflects the number of used links for the routing of all demands.

**Require:**  $G := (V, E, s, t), C, \mathcal{D}$

- 1:  $\mathbf{x} \leftarrow \mathbf{0}$
- 2: Convergence = false
- 3: **while** Convergence = false **do**
- 4:   Convergence = true
- 5:   **for**  $d \in \mathcal{D}$  **do**
- 6:      $E^- \leftarrow \{e \in E : y_e(\mathbf{x}_{-d}) > 0\}, E^+ \leftarrow E \setminus E^-$
- 7:      $c_e \leftarrow C - y_e(\mathbf{x}_{-d}), \forall e \in E$
- 8:      $\mathbf{x}_d^* \leftarrow \mathbf{0}$
- 9:     Compute  $\lambda(E^-)$  and  $x_{d,\pi}^*$  for all  $\pi \subset E^-$  by solving (Max-Flow)
- 10:      $n = \left\lceil \frac{\lambda_d - \lambda(E^-)}{C} \right\rceil$
- 11:     **for**  $k = 1 \dots n$  **do**
- 12:       Choose arbitrarily edge  $\ell \in E^+$  such that  $s(\ell) = s_d$  and  $t(\ell) = t_d$
- 13:        $x_{d,\{\ell\}}^* = (\lambda_d - \lambda(E^-))/n$
- 14:        $E^+ \leftarrow E^+ \setminus \{\ell\}, E^- \leftarrow E^- \cup \{\ell\}$
- 15:     **end for**
- 16:     **if**  $n < v(\mathbf{x}) - v(\mathbf{x}_{-d})$  **then**
- 17:       Convergence = false
- 18:        $\mathbf{x}_d \leftarrow \mathbf{x}_d^*$
- 19:     **end if**
- 20:   **end for**
- 21: **end while**

FIGURE 2. Successive approximation algorithm.

3.3.1. *Computation Time.* To evaluate the computation time for both ILP-based algorithm and successive approximation algorithm, several topologies, with different sizes, are considered for simulations. In each simulation, the two algorithms are executed on the same topology and for the same set of demands which are randomly generated. Table 2 illustrates the obtained results for each topology size. For the last three simulations, the execution of the ILP-based algorithm is stopped before its termination because lack of sufficient RAM memory.

From these results, it can be seen clearly that the successive approximation algorithm is much more faster than the ILP-based one. A routing strategy solution can be found in a few seconds for large topology sizes.

3.3.2. *Memory Consumption.* For evaluating the memory consumption related to the two algorithms, the same set of simulations as previous section is considered here. For the last three simulations, the execution of the ILP-based algorithm is stopped before its termination because lack of sufficient RAM memory. The results of simulations are illustrated in



TABLE 2. Computation Time

Topology Size	ILP-Based Algorithm	Successive Approximation Algorithm
5 nodes	0.08s	0.008s
8 nodes	0.5s	0.01s
10 nodes	2m19s	0.014s
12 nodes	6m54s	0.2s
15 nodes	27m43s	0.3s
20 nodes	36m21s	0.6s
25 nodes	52m16s	0.8s
30 nodes	>1h	0.9s
50 nodes	>2h	1.2s
100 nodes	>5h	3.4s

TABLE 3. Memory Consumption

Topology Size	ILP-Based Algorithm	Successive Approximation Algorithm
5 nodes	80M	2M
8 nodes	170M	4M
10 nodes	240M	6M
12 nodes	825M	8M
15 nodes	1.3G	9M
20 nodes	1.8G	11M
25 nodes	3.2G	14M
30 nodes	>4G	18M
50 nodes	>4G	26M
100 nodes	>4G	40M

Table 3.

As mentioned in the Table 3, important memory consumptions are recorded when solving the routing problem by the ILP-based algorithm. However, the successive approximation algorithm gave a solution with a memory consumption of 40M for a 100-node topology.

**3.3.3. Routing Strategy Cost.** The routing strategy cost is equal to the number of used links when solving the routing problem. This value is returned by the objective function in the linear program (OPT-LP) and deduced in the successive approximation algorithm after routing all demands. To compare the quality of routing for the two algorithms, it suffices to compare their routing strategies costs. For this comparison, a simplistic topology with 10 nodes is considered. Twenty five Simulations have been performed on this network. For each simulation, both ILP-based algorithm and successive approximation algorithm try to route the same set of demands, which are randomly generated. Figure 3 represents

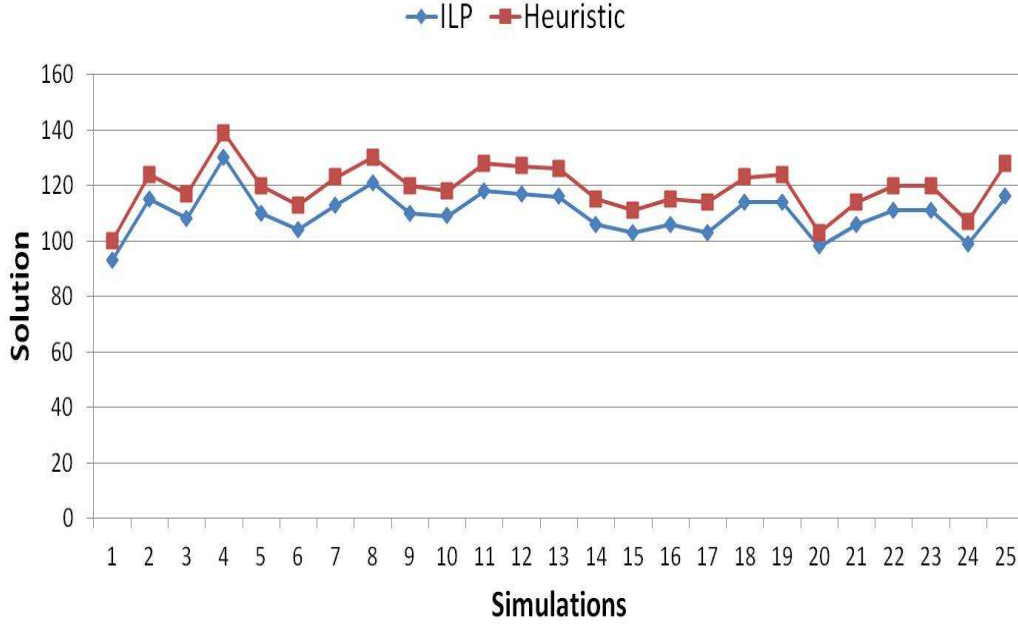


FIGURE 3. Routing Strategy Cost

two curves illustrating the routing strategy costs, recorded for the two algorithms. And for more precision, Figure 4 illustrates the relative error between the optimal and approximate solutions.

The obtained results shows that the successive approximation algorithm provides a lower quality of routing that the ILP-based algorithm, but globally acceptable. In fact, as this heuristic iterates through the demands sequentially, it attempts to find optimal solution for each demand without taking into account the impact on following ones in later iterations. Therefore, it may not always reach the performance of the ILP-based algorithm which solves the routing problem for all demands altogether. In this example shown in Figure 4, the average gap, along 25 simulations, between the two solutions is around 7,92%. Average gaps of 7,85%, 7,82% and 7,78% have been recorded, along 100 simulations, for 10-node, 15-node and 20-node topologies respectively. According to the obtained results, the routing quality ensured by the successive approximation algorithm seems satisfactory compared to the optimal ILP solution, especially if we consider the gain in time computation and memory consumption.

The average gap is expressed by

$$\frac{\sum_{simulations} \left( \frac{Sol_{approx} - Sol_{opt}}{Sol_{approx}} \right)}{nb_{simulations}}$$

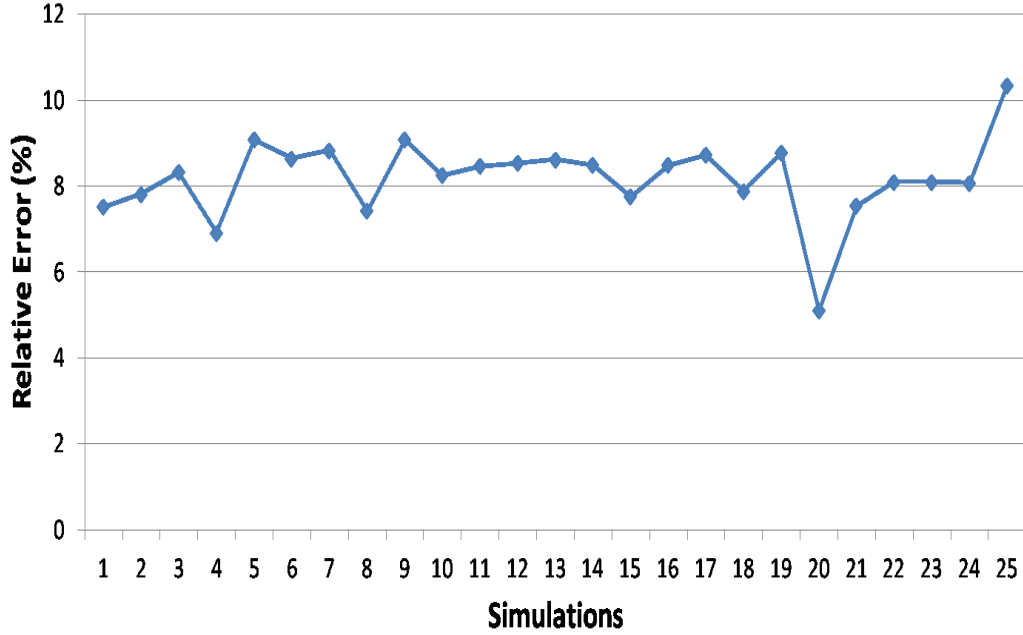


FIGURE 4. Relative error between optimal and approximate solutions

where  $Sol_{approx}$  and  $Sol_{opt}$  represent the approximative solution and the optimal solution respectively.

**3.3.4. Convergence Statistics.** An other type of statistics have been performed to evaluate the convergence of the successive approximation algorithm. These statistics consists on computing the number of iterations performed in order to converge. It represents the convergence index of the algorithm. The same set of simulations shown in the sub-section 3.3.3 are considered here. The figure 5 illustrates the obtained results.

These results confirms that the successive approximation algorithm converges in a finite number of iterations.

#### 4. ROUTING OF LIGHTPATHS AND WAVELENGTH ASSIGNMENT

The routing and wavelength assignment (RWA) problem is an optical networking problem. The general objective of the RWA problem is to maximize the number of established connections.

**4.1. Integer Linear Programming Formulation.** We consider an optical transport network represented by an directed graph  $G := (V, E)$ , where  $V$  is a set of vertices or nodes corresponding to the switches of the OTN and  $E$  is the set of optical fibers between

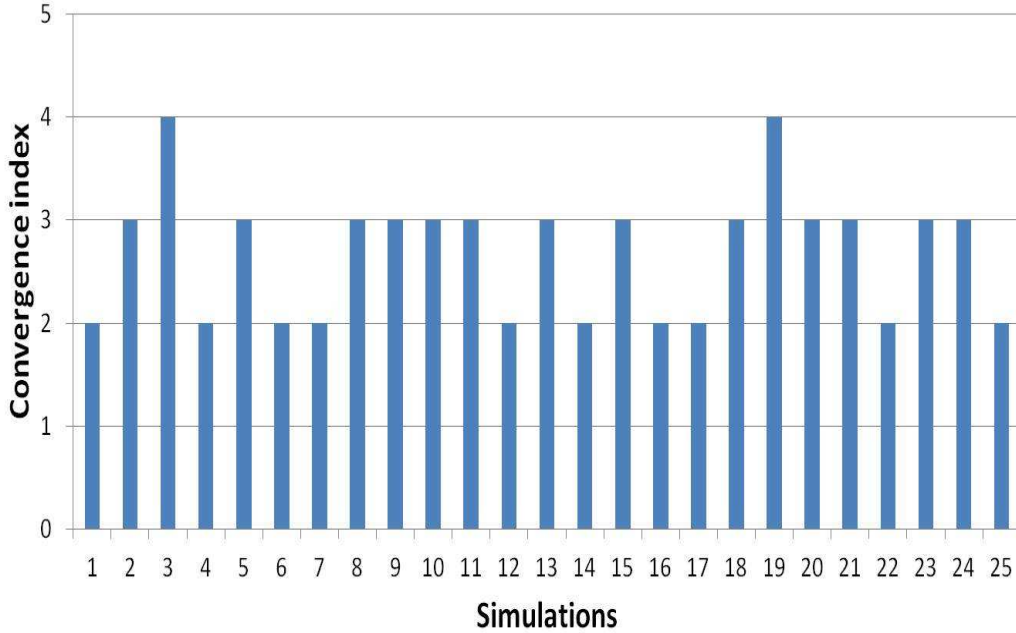


FIGURE 5. Successive Approximation Algorithm Convergence : Number of iterations

nodes. We denote by  $\mathcal{W} = \{1, \dots, W\}$  the set of wavelengths (or colours) that can be assigned to lighpaths. We are given a set  $\mathcal{K}$  of lighpaths that have to be routed in the network. Each lighpath request must be given a route and a wavelength. The wavelength must be consistent for the entire path (it is assumed that no wavelength converter is used). Two lighpaths can share the same optical link, provided a different wavelength is used. For each lighpath  $k \in \mathcal{K}$ , we let  $s_k$  be its source node and  $t_k$  be its destination node.

For each colour  $w \in \mathcal{W}$ , we define a layer of the network as a graph  $G_w = (V_w, E_w)$ , where each node of  $V_w$  is obtained by duplicating the corresponding node of  $V$ , and each edge of  $E_w$  is obtained by duplicating the corresponding directed edge in  $E$ . We thus have as many network layers as there are possible colours. This is illustrated in Figure 6. We view each network layer as a separate network where each link has capacity 1, so that a single lighpath can be routed on it. Then, the problem can be formulated as that of routing each lighpath in one and only one network layer.

For all lighpaths  $k \in \mathcal{K}$  and all colours  $w \in \mathcal{W}$ , let us define the following binary decision variables:

$$(1) \quad y_k^w = \begin{cases} 1 & \text{if lighpath } k \text{ is assigned colour } w, \\ 0 & \text{otherwise,} \end{cases}$$

and

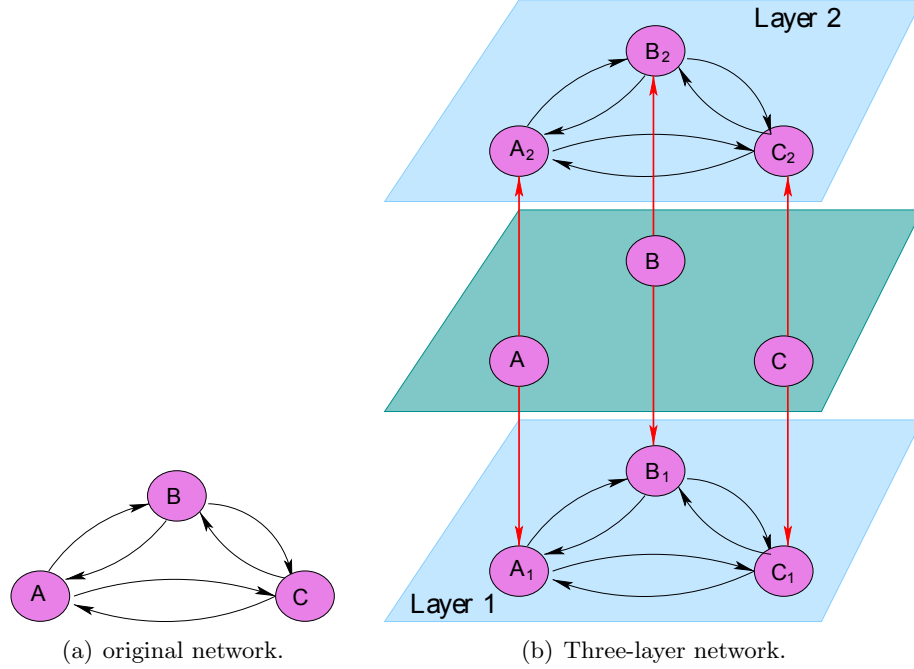


FIGURE 6. The original network and the three layers obtained by duplicating nodes and links in the case of two colours.

$$(2) \quad x_k^e = \begin{cases} 1 & \text{if lighpath } k \text{ is routed on link } e, \\ 0 & \text{otherwise,} \end{cases}$$

for all  $e \in E_W$ . Since each lighpath has to be routed in a single network layer, we clearly have

$$(3) \quad \sum_{w \in \mathcal{W}} y_k^w = 1, \quad \forall k \in \mathcal{K}.$$

Obviously, the links of network layer  $w$  are not used by lighpath  $k$  if this lighpath is not assigned the colour  $w$ , so that

$$(4) \quad x_k^e \leq y_k^w, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \forall k \in \mathcal{K}.$$

On the contrary, if lighpath  $k$  is assigned colour  $w$ , then it has to be routed in network layer  $w$ . Defining, for each node  $n \in V_w$ ,  $E_w^+(n)$  (resp.  $E_w^-(n)$ ) as the set of directed edges of  $E_w$  that are incoming (resp. outgoing) at node  $n$ , this routing problem can be expressed using a node-link formulation:

$$(5) \quad \sum_{e \in E_w^+(n)} x_k^e - \sum_{e \in E_w^-(n)} x_k^e = \begin{cases} -y_k^w & n = s_k \\ y_k^w & n = t_k \\ 0 & \text{otherwise.} \end{cases}$$

In each network layer, each link can accommodate at most one lighpath, so that we also have the following constraint:

$$(6) \quad \sum_{k \in \mathcal{K}} x_k^e \leq 1, \quad \forall e \in E_w, \forall w \in \mathcal{W}.$$

Constraints (4)-(6) define feasible solutions to the RWA problem. The goal is to minimize the number of used wavelengths, or, equivalently, the number of used network layers. For each colour  $w$ , define

$$(7) \quad u_w = \begin{cases} 1 & \text{if colour } w \text{ is used} \\ 0 & \text{otherwise.} \end{cases}$$

Since colour  $w$  is used if at least one lighpath is assigned to it, we clearly have

$$(8) \quad \sum_{k \in \mathcal{K}} y_k^w \leq K u_w, \quad \forall w \in \mathcal{W},$$

where  $K$  is the number of lighpaths. The RWA problem can now be formally defined as an integer linear program (ILP), as shown below. It has been shown that the RWA problem is NP-complete in [8]. The proof involves a reduction to the  $n$ -graph colorability problem. In other words, solving the RWA problem is as complex as finding the chromatic number of a general graph.

(RWA)

minimize  $w_{max}$ 

subject to

$$\begin{aligned}
w u_w &\leq w_{max}, \quad \forall w \in \mathcal{W}, \\
\sum_{k \in \mathcal{K}} y_k^w &\leq K u_w, \quad \forall w \in \mathcal{W}, \\
\sum_{k \in \mathcal{K}} x_k^e &\leq 1, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \\
\sum_{e \in E_w^+(s_k)} x_k^e - \sum_{e \in E_w^-(s_k)} x_k^e &= -y_k^w, \quad \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \\
\sum_{e \in E_w^+(t_k)} x_k^e - \sum_{e \in E_w^-(t_k)} x_k^e &= y_k^w, \quad \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \\
\sum_{e \in E_w^+(n)} x_k^e - \sum_{e \in E_w^-(n)} x_k^e &= 0, \quad \forall n \neq s_k, t_k, \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \\
x_k^e &\leq y_k^w, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \\
\sum_{w \in \mathcal{W}} y_k^w &= 1, \quad \forall k \in \mathcal{K}, \\
x_k^e &\in \{0, 1\}, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \\
y_k^w &\in \{0, 1\}, \quad \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \\
u_w &\in \{0, 1\}, \quad \forall w \in \mathcal{W}, \\
w_{max} &\geq 0.
\end{aligned}$$

We note the following property of optimal solutions of the above problem.

**Lemma 2.** *Let  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, w_{max})$  be an optimal solution of problem (RWA). Then*

$$(9) \quad \sum_{k \in \mathcal{K}} y_k^w \geq 1, \quad \forall w \leq w_{max},$$

*Proof.* Assume on the contrary that there exists  $q < w_{max}$  such that  $\sum_{k \in \mathcal{K}} y_k^q = 0$ . Then it is easy to see that we can define another feasible solution  $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{u}}, q)$  such that all lighpaths routed in network layer  $w_{max}$  are now routed in network layer  $q$  (with exactly the same path), while the routes of the other lighpaths are the same in both solutions. The new solution uses  $w_{max} - 1$  network layers, which implies that  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, w_{max})$  is not an optimal solution, i.e., a contradiction.  $\square$

We observe that, given a routing strategy  $\mathbf{x}$  satisfying (5)-(6), we can easily obtain a feasible solution to the above integer linear program. Indeed, if lighpath  $k$  is routed in

network layer  $w$ , then we can set  $y_k^w = 1$  and  $u_w = 1$ . This suggests that the RWA problem can be formulated as a pure routing problem.

**4.2. Formulation as an equivalent routing problem.** We shall now show that the RWA problem can be equivalently formulated as a pure routing problem. Define the following constants:

$$(10) \quad c_w = K^{w-1}, \quad w = 1, \dots, W.$$

Noting that

$$\begin{aligned} (K-1) \sum_{w=1}^q c_w &= (K-1) \sum_{n=0}^{q-1} K^n, \\ &= \sum_{n=1}^q K^n - \sum_{n=0}^{q-1} K^n, \\ &= K^q - 1, \end{aligned}$$

we see that these coefficients are such that

$$(11) \quad K \sum_{w=1}^q c_w < \sum_{w=1}^q c_w + c_{q+1}.$$

Let us now consider the following optimization problem:

$$(EQ\text{-RWA}) \quad \begin{aligned} &\text{minimize} \quad \sum_{w \in \mathcal{W}} c_w \sum_{k \in \mathcal{K}} y_k^w \\ &\text{subject to} \end{aligned}$$

$$\begin{aligned} &\sum_{k \in \mathcal{K}} x_k^e \leq 1, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \\ &\sum_{e \in E_w^+(n)} x_k^e - \sum_{e \in E_w^-(n)} x_k^e = h_k^n(y_k^w), \forall n \in V_w, \forall k \in \mathcal{K}, \forall w \in \mathcal{W}, \\ &x_k^e \leq y_k^w, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \\ &\sum_{w \in \mathcal{W}} y_k^w = 1, \quad \forall k \in \mathcal{K}, \\ &x_k^e \in \{0, 1\}, \quad \forall e \in E_w, \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \\ &y_k^w \in \{0, 1\}, \quad \forall w \in \mathcal{W}, \forall k \in \mathcal{K}, \end{aligned}$$

where we have used the following notations for  $k \in \mathcal{K}$  and  $n \in V_w$



$$(12) \quad h_k^n(y_k^w) = \begin{cases} -y_k^w & n = s_k \\ y_k^w & n = t_k \\ 0 & \text{otherwise.} \end{cases}$$

We have the following result.

**Theorem 1.** *Let  $(\mathbf{x}, \mathbf{y})$  be any optimal solution of problem (EQ-RWA). Then  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, w_{max})$  is an optimal solution of problem (RWA), where*

$$(13) \quad u_w = \min \left( 1, \sum_{k \in \mathcal{K}} y_k^w \right), \quad w = 1, \dots, W,$$

and  $w_{max} = \max_{w \in \mathcal{W}} (w u_w)$ .

*Proof.* It is clear from the definition of problem (EQ-RWA) that the vectors  $\mathbf{x}$  and  $\mathbf{y}$  satisfy equations (4)-(6). From (13), constraints (8) are also satisfied. Moreover, the definition of  $w_{max}$  implies that  $w_{max} \geq 0$  and  $w u_w \leq w_{max}$  for all  $w \in \mathcal{W}$ . Hence,  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, w_{max})$  is a feasible solution of problem (RWA).

Assume that this solution is not an optimal one, that is that there exists a feasible solution  $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{u}}, q)$  of problem (RWA) such that  $q \leq w_{max} - 1$ . This clearly implies that the vector  $\hat{\mathbf{y}}$  is such that

$$\sum_{k \in \mathcal{K}} \hat{y}_k^w = 0, \quad \forall w > q.$$

Hence,

$$\begin{aligned} \sum_{w \in \mathcal{W}} c_w \sum_{k \in \mathcal{K}} \hat{y}_k^w &= \sum_{w=1}^q c_w \sum_{k \in \mathcal{K}} \hat{y}_k^w, \\ &\leq K \sum_{w=1}^q c_w, \\ &< \sum_{w=1}^q c_w + c_{q+1}, \end{aligned}$$

where the first inequality follows from  $\sum_{k \in \mathcal{K}} \hat{y}_k^w \leq K, \forall w \in \mathcal{W}$ , and the second one follows from (11). Since  $q + 1 \leq w_{max}$  and since from Lemma 2

$$\sum_{k \in \mathcal{K}} y_k^w \geq 1, \quad \forall w \leq w_{max},$$

it yields

**Require:**  $G = (V, E)$ ,  $\mathcal{K}$  and  $\mathcal{W}$

- 1:  $w \leftarrow 1$
- 2: **while**  $\mathcal{K} \neq \emptyset$  **do**
- 3:   Solve problem (MAX-FLOW- $w$ ) for network layer  $w$ .
- 4:    $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k \in \mathcal{K} : y_k^w = 1\}$
- 5:    $w \leftarrow w + 1$
- 6: **end while**

FIGURE 7. Basic approximate algorithm (BA-RWA) to solve optimization problem (EQ-RWA).

$$\begin{aligned} \sum_{w \in \mathcal{W}} c_w \sum_{k \in \mathcal{K}} \hat{y}_k^w &< \sum_{w=1}^q c_w \sum_{k \in \mathcal{K}} y_k^w + c_{q+1} \sum_{k \in \mathcal{K}} y_k^{q+1}, \\ &< \sum_{w \in \mathcal{W}} c_w \sum_{k \in \mathcal{K}} y_k^w. \end{aligned}$$

We thus conclude that  $(\mathbf{x}, \mathbf{y})$  is not an optimal solution of problem (EQ-RWA), i.e., a contradiction. Therefore,  $(\mathbf{x}, \mathbf{y}, \mathbf{u}, w_{max})$  is an optimal solution of problem (RWA).  $\square$

According to Theorem 1, we can easily obtain an optimal solution of the original problem from an optimal solution of problem (EQ-RWA). Thus, in the following we shall study problem (EQ-RWA) instead of studying problem (RWA).

**4.3. Solution procedure.** For a given network layer  $w$ , let us define the following optimization problem:

$$\begin{aligned} \text{(MAX-FLOW-}w\text{)} \quad & \text{maximize } \sum_{k \in \mathcal{K}} y_k^w \\ & \text{subject to} \\ & \sum_{k \in \mathcal{K}} x_k^e \leq 1, \quad \forall e \in E_w, \\ & \sum_{e \in E_w^+(n)} x_k^e - \sum_{e \in E_w^-(n)} x_k^e = h_k^n(y_k^w), \forall n \in V_w, \forall k \in \mathcal{K}, \\ & x_k^e \in \{0, 1\}, \quad \forall e \in E_w, \forall k \in \mathcal{K}, \\ & y_k^w \in \{0, 1\}, \quad \forall k \in \mathcal{K}, \end{aligned}$$

In this section, we propose an heuristic for resolving the (EQ-RWA) problem. The algorithm of the proposed heuristic is described in figure 7.

In this algorithm, we seek to maximize the use of each network layer before proceeding to the next layer. The transition to the next layer is done once the current layer is saturated. Therefore, it's clear that no demand of the  $w$ -layer can be rerouted or replaced

in the overlying layers. As against, demands placed in the w-layer can be rerouted over the underlying layers, since these layers may be considered as under-used for some or all of w-layer demands. Let us note  $\mathcal{K}_w = \{k \in \mathcal{K} : y_k^w = 1\}$  the set of demands placed on the w-layer and  $L_R = \{\lambda \in \mathcal{W} : \lambda < w\}$  the set of used network layers after solving the optimization problem (EQ-RWA) with our proposed heuristic.

From this assumption, we proposed an improved algorithm based on successive approximation method. After a first routing with the algorithm proposed in figure 7, we opt to decrease the number of used network layers by trying to reroute all demands of each layer in the others underlying layers. In each w-layer iteration, we seek to find a new routing strategy where w-layer is no longer used. This algorithm try to reroute each demand  $k \in \mathcal{K}_w$  over the shortest  $\lambda$ -path where  $\lambda \in \llbracket w + 1 \dots L_R \rrbracket$ . If there is no available paths for  $k$  in all  $\lambda$ -layers, then the w-layer still used in the final routing strategy.

For a given network layer  $\lambda$ , let us define the following optimization problem for the demand  $k$ :

(SHORTEST  $\lambda$ -PATH- $k$ )

$$\begin{aligned} & \text{minimize } \sum_{e \in E_\lambda} z_e \\ & \text{subject to} \\ & \sum_{e \in E_\lambda^+(n)} z_e - \sum_{e \in E_\lambda^-(n)} z_e = \begin{cases} -1 & \text{if } n = s_k \\ 1 & \text{if } n = t_k \\ 0 & \text{otherwise.} \end{cases}, \quad \forall n \in V_\lambda, \\ & z_e \leq \delta_e, \quad \forall e \in E_\lambda, \\ & z_e \in \{0, 1\}, \quad \forall e \in E_\lambda. \end{aligned}$$

where we have used the following notations for  $e \in E_\lambda$

$$(14) \quad \delta_e = \begin{cases} 1 & \text{if } e \text{ is } \lambda\text{-available} \\ 0 & \text{otherwise.} \end{cases}$$

The enhanced algorithm is described in figure 8

**4.4. Experiments and results.** In this section, several experiments on the optimization problem for traffic routing were performed in order to evaluate and validate the effectiveness of the proposed solutions. The GUROBI solver is used for solving the two integer linear programs: (RWA) and (EQ-RWA). The aim of the experiments here is to analyze the performance of algorithms and compare the proposed methods. The comparison is performed between the different proposed algorithms: ILP-based algorithm (RWA), equivalent ILP-based algorithm (EQ-RWA), basic and enhanced algorithms to solve optimization problem (EQ-RWA) which are illustrated in Fig. 7 and Fig. 8 respectively.

**Require:**  $G = (V, E)$ ,  $\mathcal{K}$  and  $\mathcal{W}$

```

1:  $w \leftarrow 1$ 
2: while  $\mathcal{K} \neq \emptyset$  do
3:   Solve problem (MAX-FLOW- $w$ ) for network layer  $w$ .
4:    $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k \in \mathcal{K} : y_k^w = 1\}$ 
5:    $w \leftarrow w + 1$ 
6: end while
7:  $L_R \leftarrow \{\lambda \in \mathcal{W} : \lambda < w\}$ 
8: for  $w \in L_R$  do
9:    $\mathcal{K}_w \leftarrow \{k \in \mathcal{K} : y_k^w = 1\}$ 
10:   $\mathcal{K}_{w-tmp} \leftarrow \mathcal{K}_w$ 
11: end for
12:  $L_A \leftarrow \emptyset$ 
13:  $amelioration \leftarrow 0$ 
14: for  $w \in L_R$  do
15:    $cpt \leftarrow 0$ 
16:   for  $k \in \mathcal{K}_w$  do
17:      $rerouted \leftarrow false$ 
18:     for  $\lambda = w + 1 \dots |L_R|$  do
19:       Solve problem (SHORTEST  $\lambda$ -PATH- $k$ ) for network layer  $\lambda$ .
20:       if  $\{e \in E_\lambda : z_e = 1\} \neq \emptyset$  then
21:          $\mathcal{K}_{\lambda-tmp} \leftarrow \mathcal{K}_{\lambda-tmp} \cup \{k\}$ ,  $\mathcal{K}_{w-tmp} \leftarrow \mathcal{K}_{w-tmp} \setminus \{k\}$ 
22:          $rerouted \leftarrow true$ ,  $cpt \leftarrow cpt + 1$ 
23:         break
24:       end if
25:     end for
26:     if  $rerouted = false$  then
27:       for  $\lambda = w \dots |L_R|$  do
28:          $\mathcal{K}_{\lambda-tmp} \leftarrow \mathcal{K}_\lambda$ 
29:       end for
30:     break
31:     end if
32:   end for
33:   if  $cpt = |\mathcal{K}_w|$  then
34:      $L_A \leftarrow L_A \cup \{w\}$ 
35:     for  $\lambda = w \dots |L_R|$  do
36:        $\mathcal{K}_\lambda \leftarrow \mathcal{K}_{\lambda-tmp}$ 
37:     end for
38:   end if
39: end for
40:  $L_R \leftarrow L_R \setminus L_A$ 
41:  $amelioration \leftarrow |L_A|$ 

```

FIGURE 8. Enhanced approximate algorithm (EA-RWA) to solve optimization problem (EQ-RWA).

TABLE 4. Computation Time for RWA proposed algorithms

Topology Size	(RWA)	(EQ-RWA)	(BA-RWA)	(EA-RWA)
$ V  = 5,  E  = 12,  W  = 20$	0.5s	0.5s	0.2s	0.3s
$ V  = 8,  E  = 20,  W  = 20$	10s	1m5s	1.6s	1.8s
$ V  = 10,  E  = 26,  W  = 30$	2m19s	2m43s	4.2s	4.5s
$ V  = 15,  E  = 40,  W  = 40$	6m52s	8m24s	12.2s	12.8s
$ V  = 20,  E  = 56,  W  = 40$	13m12s	17m05s	52.2s	53s
$ V  = 30,  E  = 86,  W  = 60$	38m56s	49m21s	4m12s	4m14s
$ V  = 40,  E  = 122,  W  = 60$	>2h	>2h	9m34s	9m38s
$ V  = 50,  E  = 158,  W  = 80$	>3h	>3h	27m42s	27m53s

TABLE 5. Memory Consumption for RWA proposed algorithms

Topology Size	(RWA)	(EQ-RWA)	(BA-RWA)	(EA-RWA)
$ V  = 5,  E  = 12,  W  = 20$	5M	6M	5M	6M
$ V  = 8,  E  = 20,  W  = 20$	80M	85M	9M	11M
$ V  = 10,  E  = 26,  W  = 30$	280M	295M	14M	16M
$ V  = 15,  E  = 40,  W  = 40$	520M	545M	18M	20M
$ V  = 20,  E  = 56,  W  = 40$	940M	980M	26M	28M
$ V  = 30,  E  = 86,  W  = 60$	1.9G	2G	40M	43M
$ V  = 40,  E  = 122,  W  = 60$	2.8G	>3G	75M	79M
$ V  = 50,  E  = 158,  W  = 80$	>3G	>3G	200M	206M

The performance comparison between the algorithms is done based on two criteria. The first is related to the execution performance including computation time and memory consumption. The second one concerns the routing strategy cost which represents in this case the number of used wavelengths (i.e. colours) for satisfying all routed lightpaths.

4.4.1. *Execution performance.* In this part, we summarize the run performance of the proposed algorithms on several topologies, with different sizes. The results related to computation time and memory consumption are illustrated in Tables 4 and 5, respectively. For each simulation, the different algorithms are executed on the same topology and for the same set of demands which are randomly generated. Topology size is defined by three terms  $|V|$ ,  $|E|$  and  $|W|$  which represents the number of nodes, the number of fibers and the number of available wavelengths in each fiber, respectively.

From these results, it's clear that the approximate algorithms are much more faster, and consume less memory than the ILP-based ones. A routing strategy solution can be found in a few seconds for large topology sizes.

4.4.2. *Routing strategy cost.* In RWA problem, the routing strategy cost represents the number of used layers (i.e. wavelengths) when routing the lightpaths matrix. This optimization criterion is the same in all our proposed algorithms. In this part, we compare the

TABLE 6. Routing strategy cost

Matrix size	(EQ-RWA)	(BA-RWA)	(EA-RWA)
89	13 $\lambda$	14 $\lambda$	13 $\lambda$
86	14 $\lambda$	15 $\lambda$	14 $\lambda$
104	16 $\lambda$	16 $\lambda$	16 $\lambda$
97	14 $\lambda$	14 $\lambda$	14 $\lambda$
80	13 $\lambda$	14 $\lambda$	13 $\lambda$
81	14 $\lambda$	15 $\lambda$	14 $\lambda$
82	16 $\lambda$	16 $\lambda$	16 $\lambda$
98	15 $\lambda$	16 $\lambda$	15 $\lambda$
83	14 $\lambda$	14 $\lambda$	14 $\lambda$
92	13 $\lambda$	14 $\lambda$	14 $\lambda$
169	25 $\lambda$	25 $\lambda$	25 $\lambda$
174	28 $\lambda$	28 $\lambda$	28 $\lambda$
187	27 $\lambda$	28 $\lambda$	28 $\lambda$
158	22 $\lambda$	23 $\lambda$	22 $\lambda$
174	26 $\lambda$	27 $\lambda$	26 $\lambda$
168	27 $\lambda$	27 $\lambda$	27 $\lambda$
172	27 $\lambda$	27 $\lambda$	27 $\lambda$
165	23 $\lambda$	24 $\lambda$	24 $\lambda$
169	28 $\lambda$	29 $\lambda$	29 $\lambda$
177	26 $\lambda$	27 $\lambda$	27 $\lambda$

quality of routing ensured by the different solutions. For that, it suffices to compare their routing strategy costs. For this comparison, a simplistic topology with 10 nodes, 26 fibers and 40 available wavelengths is considered. Twenty simulations have been performed on this network. For each simulation, the different algorithms try to route the same set of demands, which are randomly generated. Table 6 illustrates the obtained results and figure 9 represents the curve of the relative error between the ILP-based algorithm (EQ-RWA) solution (i.e. optimal solution) and the (EA-RWA) one (i.e. approximate solution), along the 20 simulations. The matrix size here represents the total number of lightpath demands.

The obtained results clearly show the efficiency of the proposed enhanced heuristic. Among the twenty simulations, there are only five times where we don't reach the optimal solution. In addition, the (EA-RWA) algorithm provides an improved solution compared to the (BA-RWA) algorithm. In several simulations, we succeeded in enhancing the strategy routing cost returned by the (BA-RWA) thanks to our enhanced algorithm.

#### 4.5. New heuristic as an equivalent to "Maximum Multicommodity Flow" to solve the (MAX-FLOW- $w$ ) problem.

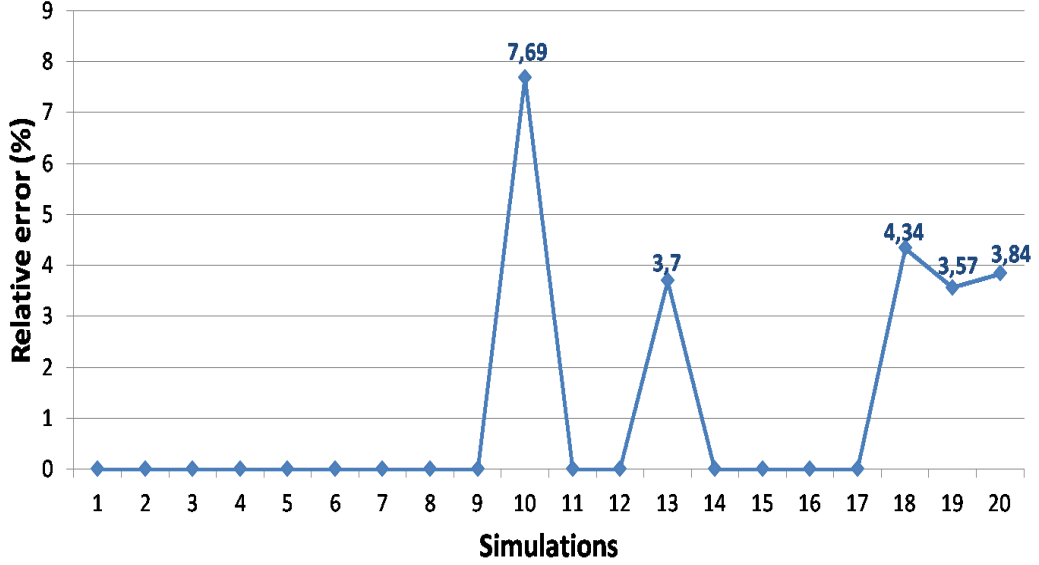


FIGURE 9. Relative error between approximate and optimal solutions

4.5.1. *Adopted Approach.* As shown in the previous sections, our approximation algorithm (BA-RWA) is based on the linear program (MAX-FLOW- $w$ ). This program try to route the maximum number of demands (i.e. lightpaths) over the  $w$ -layer. It ensures an optimal solution per network  $w$ -layer, but it can be time-limited and memory-limited for resolving the RWA problem over real networks. For this, we have opted to finding an efficient heuristic that solves the same (MAX-FLOW- $w$ ) problem.

Since we search to maximize the number of routed lightpaths on the  $w$ -layer given that each link (i.e. fiber) can be used only once, the problem is equivalent to a Maximum Multicommodity Problem with edges capacities equal to 1. Commodity pairs represents the set of pairs  $(s_k, t_k)$  of lightpaths for  $k \in \mathcal{K}$ .

A multicommodity flow problem is defined on a directed network  $G = (V, E)$  with capacities  $u(e)$  for  $e \in E$  and  $p$  source-target pairs  $(s_j, t_j)$  for  $1 \leq j \leq p$ . The problem is to find flows  $f_j$  from  $s_j$  to  $t_j$  that satisfy flow conservation constraints and capacity constraints that ensures that the sum of flows on any edge does not exceed the capacity of the edge. For the maximum multicommodity flow problem, the objective is to maximize the sum of the flows  $\sum_j |f_j|$ .

While there are many different algorithms known for this problem we discuss one which guarantee an  $(1 + \omega)$ -approximation to the maximum throughput. The goal of the Garg-Knemann algorithm is to find an  $\omega$ -approximate solution for any error parameter  $\omega > 0$ . the  $\omega$ -approximate solution is a flow that has value at least  $(1 - \omega)$  times the maximum

**Require:** network  $G = (V, E)$ , capacities  $u(e)$ , commodity pairs  $(s_j, t_j)$  for  $1 \leq j \leq p$ , accuracy  $\epsilon$

- 1: Initialize  $l(e) = \phi \quad \forall e, x = 0$
- 2: **while** there is a  $P \in \mathcal{P}$  with  $\sum_{e \in P} l(e) < 1$  **do**
- 3:   Select the minimum length path  $P \in \mathcal{P}$
- 4:    $u \leftarrow \min_{e \in P} u(e)$
- 5:    $x(P) \leftarrow x(P) + u$
- 6:    $\forall e \in P, \quad l(e) \leftarrow l(e)(1 + \frac{\epsilon u}{u(e)})$
- 7: **end while**
- 8: Return  $(x, l)$ .

FIGURE 10. Generic Garg-Knemmann algorithm.

value. In addition, the running time of this type of algorithms is known depending on the approximation value  $\omega$ , the number of commodity pairs  $p$  and the number of edges  $m$ .

This generic algorithm for maximum multicommodity flow, that we adopted here, try to find mutually two solutions for the primal and dual programs for the maximum multicommodity flow problem. Let  $\mathcal{P}_j$  denote the set of paths from  $s_j$  to  $t_j$ , and  $\mathcal{P} := \cup_j \mathcal{P}_j$ .  $x(P)$  equals the amount of flow sent along path  $P$  and represents the decision variable of the primal program. The length of an edge  $l(e)$  represents the decision variable of the dual program and reflects the marginal cost of using an additional unit of capacity of the edge. The generic Garg-Knemmann algorithm, described in figure 10, stops when the ratio between the primal and the dual solutions is at most  $1 + \omega$ . The accuracy  $\epsilon$  is a fixed constant that chosen appropriately depending on  $\omega$ .  $\phi$  represents the inial length of all network edges which depends on  $\epsilon$  and  $m$ .

The Garg-Knemmann algorithm solves the maximum multicommodity flow problem with a guarantee of error approximation. The real flow can be deducted by scaling the final flow obtained in the generic algorithm by  $\log_{1+\epsilon} \frac{1+\epsilon}{\phi}$ . But the generated flow is generally fractional which is not adapted in our study case where we search to route a set of lightpaths (i.e. flows in the case of maximum multicommodity flow problem). Therefore, the solution must be integer or at least quasi-integer. For this, we have proposed a modification in Garg-Knemmann algorithm in order to converge to a quasi-integer solution: Since in each iteration we search the minimum length path and the algorithm augments flow along this path, so we can use different paths in two successive iterations. Our idea consists on keeping the chosen minimum length path and augmenting flow along it until the "real" flow converges to 1. Once this condition is satisfied, the algorithm can switch to another minimum length path and repeat the same treatment. The modified Garg-Knemmann algorithm is described in figure 11. The proposed modification allows a primal quasi-integer solution. This solution can be easily rounded to an integer solution.

The Garg-Knemmann algorithm try to maximize the flow sent between the source and the taget of each commodity pair. Hence, the calculated flow for a commodity  $j$  can exceed the required amount of demand (i.e. number of lightpaths) between  $s_j$  and  $t_j$ . This does



**Require:** network  $G = (V, E)$ , capacities  $u(e)$ , commodity pairs  $(s_j, t_j)$  for  $1 \leq j \leq p$ , amount of demands  $t_j$  for  $1 \leq j \leq p$ , accuracy  $\epsilon$ , threshold  $\alpha$  (close to 1)

- 1: Initialize  $l(e) = \phi \quad \forall e, x = 0$
- 2:  $iter \leftarrow 1$
- 3: **while** there are a commodity  $j \in \llbracket 1 \dots p \rrbracket$  and a path  $P \in \mathcal{P}_j : \sum_{e \in P} l(e) < 1$  **and**  $\sum_{Q \in \mathcal{P}_j} x(Q) < t_j$  **do**
- 4:   **if**  $iter = 1$  **or**  $x(P) \geq \alpha$  **or**  $\sum_{Q \in \mathcal{P}_j} x(Q) \geq t_j$  **then**
- 5:     Select the minimum length path  $P \in \mathcal{P}$  and the commodity  $j : P \in \mathcal{P}_j$  **and**  $\sum_{Q \in \mathcal{P}_j} x(Q) < t_j$
- 6:   **end if**
- 7:    $u \leftarrow \min_{e \in P} u(e)$
- 8:    $x(P) \leftarrow x(P) + \frac{u}{\log_{1+\epsilon} \frac{1+\epsilon}{\phi}}$
- 9:    $\forall e \in P, \quad l(e) \leftarrow l(e) \left(1 + \frac{\epsilon u}{u(e)}\right)$
- 10:  $iter \leftarrow iter + 1$
- 11: **end while**
- 12: Return  $(x, l)$ .

FIGURE 11. Modified Garg-Knemmann algorithm.

not match with our expectation. Consequently, we have added an additional capacity constraints for commodity pairs. For each commodity  $j$ , we have assigned a constant  $t_j$  which is equal to the required demand between  $s_j$  and  $t_j$ . The total flow for commodity  $j$  ( $= \sum_{Q \in \mathcal{P}_j} x(Q)$ ) must not exceed the value of  $t_j$ . Adopting this modification, we can guarantee that the maximum flow sent between  $s_j$  and  $t_j$  can not exceed the required number of lightpaths for the commodity  $j$ .

After applying our updates, we can adopt this modified algorithm for solving the (MAX-FLOW- $w$ ) problem.

4.5.2. *Experiments and results.* several experiments on the optimization problem (MAX-FLOW- $w$ ) were performed in order to evaluate and validate the effectiveness of the proposed solutions. The GUROBI solver is used for solving the integer linear program (MAX-FLOW- $w$ ). The aim of the experiments here is to analyze the performance of algorithms and compare the proposed methods. The comparison is performed between the ILP-based algorithm and the modified Garg-Knemmann algorithm after its adaptation to our optimization case. We compare here the optimal solution generated by the ILP-based algorithm and the approximate solution deduced by the modified Garg-Knemmann algorithm. For this comparison, three simplistic topologies are considered. Ten simulations have been performed on each network. For each simulation, the two algorithms try to route the same set of demands (i.e. lightpaths), which are randomly generated. Tables 7, 8 and 9 illustrate the obtained results for the three studied topologies. the fixed  $\omega$  for the Garg-Knemmann algorithm is 0,05.

TABLE 7. Topology 1 : 6 nodes and 14 links

Number of demands	Optimal solution	Approximate solution (shortest-path-first)	Approximate solution (Modified Garg-Knemann)	Rounded solution (Modified Garg-Knemann)
40	13 <b>0.4s</b>	13 <b>0.009s</b>	12,9247 <b>0.04s</b>	13
42	9 <b>0.4s</b>	9 <b>0.008s</b>	8,9894 <b>0.02s</b>	9
48	10 <b>0.5s</b>	10 <b>0.007s</b>	9,9893 <b>0.02s</b>	10
41	12 <b>0.4s</b>	12 <b>0.008s</b>	11,9707 <b>0.03s</b>	12
39	9 <b>0.4s</b>	9 <b>0.007s</b>	8,9986 <b>0.02s</b>	9
49	10 <b>0.6s</b>	10 <b>0.009s</b>	9,9985 <b>0.03s</b>	10
34	10 <b>0.4s</b>	10 <b>0.007s</b>	9,9618 <b>0.02s</b>	10
55	11 <b>0.5s</b>	11 <b>0.008s</b>	10,9892 <b>0.02s</b>	11
47	11 <b>0.4s</b>	11 <b>0.007s</b>	10,9617 <b>0.02s</b>	11
42	12 <b>0.4s</b>	12 <b>0.008s</b>	11,9707 <b>0.03s</b>	12

#### 4.6. New heuristic as an equivalent to "Maximum Edge-Dijoint Paths" to solve the (MAX-FLOW- $w$ ) problem.

4.6.1. *Adopted Algorithm.* As shown in the previous section, we can solve the (MAX-FLOW- $w$ ) problem effectively using a modified version of the Maximum Multicommodity Flow Garg-Knemann algorithm. the adopted algorithm try to route the maximum amount of flows over a network  $w$ -layer. The obtained solution is generally fractionnal, but we can easily provide a rounded feasible solution.

Since a flow unit represents a lightpath and a flow unit path can be used only once because the capacity of every edge in  $w$ -layer is 1, the problem can be distinguish as a "Maximum Edge-Dijoint Paths" (MEDP). The goal is to route as many requests as possible along edge-dijoint paths. The input consists of a directed graph  $G = (V, E)$  and a set containing  $k$  requests  $\mathcal{K} = \{(s_i, t_i) | i = 1 \dots k\}$ , where each request is a pair of vertices in  $V$ . The request  $(s_i, t_i)$  asks for a directed path from  $s_i$  to  $t_i$  in  $G$ . A feasible solution is given by a subset  $\mathcal{A}$  and an assignment of edge-disjoint paths to all requests in that subset. More precisely, each  $(s_i, t_i) \in \mathcal{A}$  must be assigned a directed path  $P_i$  from  $s_i$  to  $t_i$  in  $G$  such that no two paths  $P_i$  and  $P_j$  (for  $i, j \in \mathcal{A}$  and  $i \neq j$ ) have a directed edge of the graph

TABLE 8. Topoogy 2 : 15 nodes and 38 links

Number of demands	Optimal solution	Approximate solution (shortest-path-first)	Approximate solution (Modified Garg-Knemann)	Rounded solution (Modified Garg-Knemann)
263	27 1.5s	27 0.02s	26,9701 0.1s	27
308	31 1.3s	31 0.026s	30,9555 0.2s	31
328	31 1.3s	31 0.027s	30,9492 0.2s	31
315	28 1.3s	27 0.021s	26,9618 0.1s	27
289	31 1.4s	31 0.026s	30,9683 0.2s	31
316	30 1.3s	29 0.024s	28,9971 0.1s	29
312	30 1.3s	30 0.024s	29,9349 0.1s	30
304	34 1.3s	34 0.026s	33,9031 0.2s	34
295	31 1.5s	31 0.024s	30,9459 0.1s	31
211	26 1.3s	26 0.021s	25,9602 0.1s	26

in common. The requests in  $\mathcal{A}$  are called the accepted requests. We search to maximize the cardinality of  $\mathcal{A}$ .

The MEDP problem has been studied in many works and solved by several algorithms. One of these algorithms is the Shortest-Path-First suggested by Kolliopoulos and Stein. The algorithm is illustrated in figure 12

**4.7. Comparison between (RWA) solution and approximate solution using the new heuristic.** In this section, we compare the performance of the three algorithms : the (RWA) ILP-based algorithm, the (BA-RWA) algorithm and the (EA-RWA) algorithm. For the two last algorithms, illustrated in figure 7 and 8, the (MAX-FLOW- $w$ ) ILP-based algorithm is replaced by our new heuristic "Modified Garg-Knemann algorithm" shown if figure 11. We adopt here the same network model (10 nodes, 26 links and 40 available wavelengths) and the same set of simulations we used in the section 4.4.2. We recall that we compare the routing strategy costs for the three algorithms. The routing strategy cost represents the number of used layers (i.e. wavelengths) for routing all lightpath demands. Table 10 represents the results among twenty simulations and figure 13 show the relative error between the optimal solution and the approximate (EA-RWA) solution.

TABLE 9. Topoogy 3 : 34 nodes and 92 links

Number of demands	Optimal solution	Approximate solution (shortest-path-first)	Approximate solution (Modified Garg-Knemann)	Rounded solution (Modified Garg-Knemann)
1128	76 5.4s	76 0.208s	75,8519 2.3s	76
1119	72 6.2s	72 0.202s	71,8396 2.1s	72
1091	70 5.8s	70 0.193s	69,8849 1.9s	70
1089	71 6.3s	71 0.194s	70,8561 1.9s	71
1143	73 5.6s	72 0.209s	71,8837 2.1s	72
1103	71 6.1s	70 0.196s	69,8503 1.9s	70
1126	70 5.5s	70 0.194s	69,8529 1.9s	70
1108	73 6.4s	73 0.207s	72,8785 2.1s	73
1116	67 5.2s	66 0.186s	65,8944 1.8s	66
1058	65 5.1s	63 0.177s	62,8864 1.5s	63

**Require:** network  $G = (V, E)$ ,  $\mathcal{K} = \{(s_i, t_i) | i = 1 \dots k\}$

- 1:  $\mathcal{A} \leftarrow \emptyset$
- 2: **while**  $\mathcal{K}$  contains a request that can be routed in  $G$  **do**
- 3:   Select a request  $(s_i, t_i)$  in  $\mathcal{K}$  such that the shortest path from  $s_i$  to  $t_i$  in  $G$  has minimum length among all requests in  $\mathcal{K}$
- 4:    $\mathcal{A} \leftarrow \mathcal{A} \cup \{(s_i, t_i)\}$
- 5:    $\mathcal{K} \leftarrow \mathcal{K} \setminus \{(s_i, t_i)\}$
- 6:    $P_i \leftarrow$  a shortest path from  $s_i$  to  $t_i$  in  $G$
- 7:   Remove all edges of  $P_i$  from  $G$
- 8: **end while**
- 9: Return  $\mathcal{A}$  and  $\{P_i | (s_i, t_i) \in \mathcal{A}\}$ .

FIGURE 12. Shortest-Path-First algorithm.

Computation time and memory consumption are also compared for the three studied algorithms. we summarize here the run performance of the proposed algorithms on several topologies, with different sizes. The results related to computation time and memory

TABLE 10. Routing strategy cost : Comparison between optimal solution and approximate ones

Number of lightpaths	(RWA) or (EQ-RWA)	(BA-RWA) $\rightarrow$ (EA-RWA) (based on Modified Garg-Knemann)	(BA-RWA) $\rightarrow$ (EA-RWA) (based on Shortest-path-first)
89	13 $\lambda$	13 $\lambda \rightarrow$ 13 $\lambda$	13 $\lambda \rightarrow$ 13 $\lambda$
86	14 $\lambda$	15 $\lambda \rightarrow$ 15 $\lambda$	15 $\lambda \rightarrow$ 15 $\lambda$
104	16 $\lambda$	17 $\lambda \rightarrow$ 16 $\lambda$	17 $\lambda \rightarrow$ 16 $\lambda$
97	14 $\lambda$	14 $\lambda \rightarrow$ 14 $\lambda$	14 $\lambda \rightarrow$ 14 $\lambda$
80	13 $\lambda$	14 $\lambda \rightarrow$ 13 $\lambda$	14 $\lambda \rightarrow$ 14 $\lambda$
81	14 $\lambda$	15 $\lambda \rightarrow$ 14 $\lambda$	15 $\lambda \rightarrow$ 14 $\lambda$
82	16 $\lambda$	16 $\lambda \rightarrow$ 16 $\lambda$	16 $\lambda \rightarrow$ 16 $\lambda$
98	15 $\lambda$	16 $\lambda \rightarrow$ 16 $\lambda$	16 $\lambda \rightarrow$ 16 $\lambda$
83	14 $\lambda$	15 $\lambda \rightarrow$ 14 $\lambda$	15 $\lambda \rightarrow$ 14 $\lambda$
92	13 $\lambda$	14 $\lambda \rightarrow$ 14 $\lambda$	14 $\lambda \rightarrow$ 14 $\lambda$
169	25 $\lambda$	27 $\lambda \rightarrow$ 25 $\lambda$	27 $\lambda \rightarrow$ 25 $\lambda$
174	28 $\lambda$	30 $\lambda \rightarrow$ 29 $\lambda$	30 $\lambda \rightarrow$ 29 $\lambda$
187	27 $\lambda$	29 $\lambda \rightarrow$ 29 $\lambda$	29 $\lambda \rightarrow$ 29 $\lambda$
158	22 $\lambda$	23 $\lambda \rightarrow$ 23 $\lambda$	24 $\lambda \rightarrow$ 23 $\lambda$
174	26 $\lambda$	29 $\lambda \rightarrow$ 28 $\lambda$	29 $\lambda \rightarrow$ 28 $\lambda$
168	27 $\lambda$	27 $\lambda \rightarrow$ 27 $\lambda$	27 $\lambda \rightarrow$ 27 $\lambda$
172	27 $\lambda$	28 $\lambda \rightarrow$ 28 $\lambda$	28 $\lambda \rightarrow$ 28 $\lambda$
165	23 $\lambda$	25 $\lambda \rightarrow$ 24 $\lambda$	25 $\lambda \rightarrow$ 24 $\lambda$
169	28 $\lambda$	30 $\lambda \rightarrow$ 30 $\lambda$	30 $\lambda \rightarrow$ 30 $\lambda$
177	26 $\lambda$	28 $\lambda \rightarrow$ 27 $\lambda$	28 $\lambda \rightarrow$ 27 $\lambda$

consumption are illustrated in Tables 11 and 12, respectively. For each simulation, the different algorithms are executed on the same topology and for the same set of demands which are randomly generated. Topology size is defined by three terms  $|V|$ ,  $|E|$  and  $|W|$  which represents the number of nodes, the number of fibers and the number of available wavelengths in each fiber, respectively.

#### REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.
- [2] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *Selected Areas in Communications, IEEE Journal on*, 14(5):903–908, 1996.
- [3] S. Baroni, P. Bayvel, and R. J. Gibbens. On the number of wavelengths in arbitrarily-connected wavelength-routed optical networks. *University of Cambridge, Statistical Laboratory Research Report 1998-7*, <http://www.statslab.cam.ac.uk/reports/1998/1998-7.pdf>, 1998.
- [4] R. Barry and S. Subramaniam. The max sum wavelength assignment algorithm for wdm ring networks. In *Optical Fiber Communication. OFC 97., Conference on*, pages 121–122. IEEE, 1997.

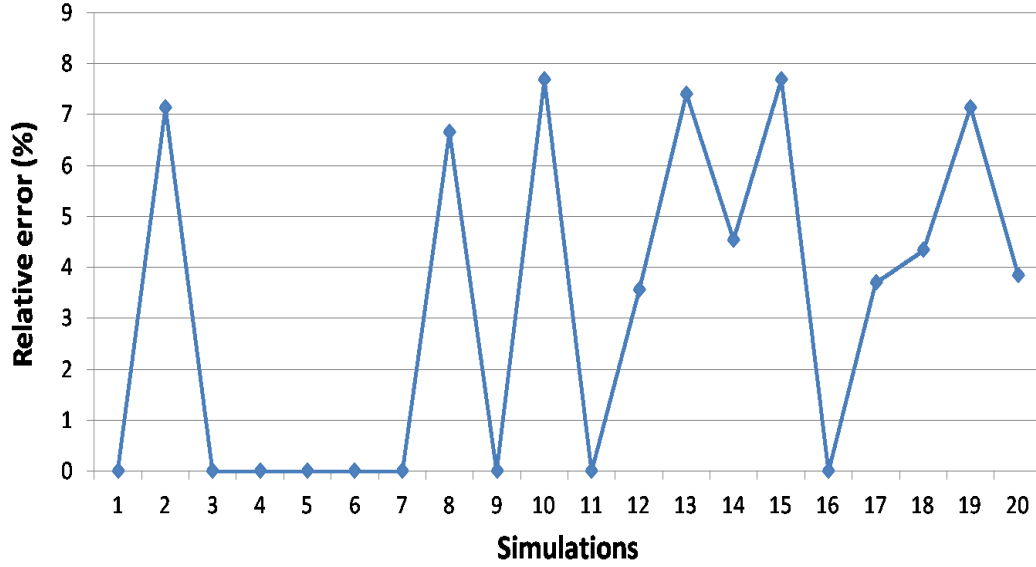


FIGURE 13. Relative error between optimal and (EA-RWA) approximate (based on Modified Garg-Knemann algorithm) solutions

TABLE 11. Computation Time : Comparison between ILP-based algorithm and proposed heuristics

Topology Size	(RWA)	(BA-RWA) → (EA-RWA) (based on Modified Garg-Knemann)	(BA-RWA) → (EA-RWA) (based on Shortest-path-first)
$ V  = 5,  E  = 12,  W  = 20$	0.5s	0.04s → 0.06s	0.002s → 0.003s
$ V  = 8,  E  = 20,  W  = 20$	10s	0.2s → 0.23s	0.023s → 0.026s
$ V  = 10,  E  = 26,  W  = 30$	2m19s	0.4 → 0.45s	0.042s → 0.046s
$ V  = 15,  E  = 40,  W  = 40$	6m52s	0.9s → 1.05s	0.09s → 0.1s
$ V  = 20,  E  = 56,  W  = 40$	13m12s	2.1s → 2.3s	0.25s → 0.27s
$ V  = 30,  E  = 86,  W  = 60$	38m56s	8.5s → 8.8s	0.92s → 0.98s
$ V  = 40,  E  = 122,  W  = 60$	>2h	17.8s → 18.3s	1.86s → 1.92s
$ V  = 50,  E  = 158,  W  = 80$	>3h	39.4s → 40.2s	3.76s → 4.38s

- [5] D. Cavendish, A. Kolarov, and B. Sengupta. Routing and wavelength assignment in wdm mesh networks. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 2, pages 1016–1022. IEEE, 2004.
- [6] B. Chen, G. N. Rouskas, and R. Dutta. On hierarchical traffic grooming in wdm networks. *IEEE/ACM Transactions on Networking (TON)*, 16(5):1226–1238, 2008.
- [7] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical wan's. *Communications, IEEE Transactions on*, 40(7):1171–1182, 1992.

TABLE 12. Memory Consumption : Comparison between ILP-based algorithm and proposed heuristics

Topology Size	(RWA)	(BA-RWA) $\rightarrow$ (EA-RWA) (based on Modified Garg-Knemann)	(BA-RWA) $\rightarrow$ (EA-RWA) (based on Shortest-path-first)
$ V  = 5,  E  = 12,  W  = 20$	5M	3M $\rightarrow$ 3.4M	2.6M $\rightarrow$ 2.9M
$ V  = 8,  E  = 20,  W  = 20$	80M	5.2M $\rightarrow$ 5.7M	4.5M $\rightarrow$ 4.9M
$ V  = 10,  E  = 26,  W  = 30$	280M	8.6M $\rightarrow$ 9.3M	8M $\rightarrow$ 8.6M
$ V  = 15,  E  = 40,  W  = 40$	520M	10M $\rightarrow$ 11.5M	9M $\rightarrow$ 10.2M
$ V  = 20,  E  = 56,  W  = 40$	940M	14M $\rightarrow$ 16M	13M $\rightarrow$ 14.6M
$ V  = 30,  E  = 86,  W  = 60$	1.9G	24M $\rightarrow$ 26M	21M $\rightarrow$ 22.8M
$ V  = 40,  E  = 122,  W  = 60$	2.8G	42M $\rightarrow$ 44M	40M $\rightarrow$ 42M
$ V  = 50,  E  = 158,  W  = 80$	>3G	85M $\rightarrow$ 88M	81M $\rightarrow$ 84M

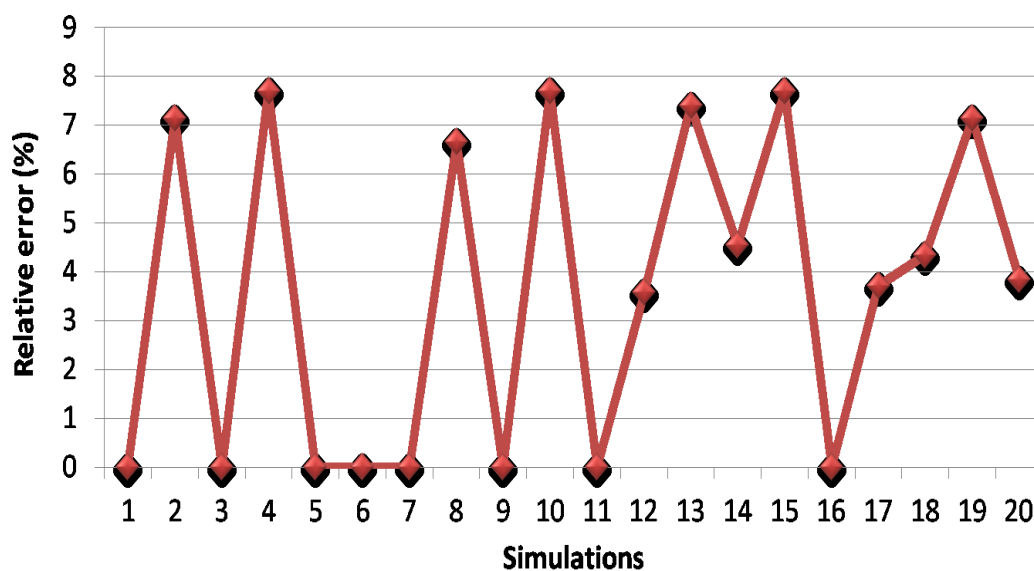


FIGURE 14. Relative error between optimal and (EA-RWA) approximate (based on shortest-path-first algorithm) solutions

- [8] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical wan's. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.
- [9] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos. Comparison of routing and wavelength assignment algorithms in wdm networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–6. IEEE, 2008.
- [10] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos. Offline routing and wavelength assignment in transparent wdm networks. *Networking, IEEE/ACM Transactions on*, 18(5):1557–1570, 2010.

- [11] P. Cochrane, R. Heckingbottom, and D. Heatley. The hidden benefits of optical transparency. *Communications Magazine, IEEE*, 32(9):90–97, 1994.
- [12] A. Ganz and X. Wang. Efficient algorithm for virtual topology design in multihop lightwave networks. *Networking, IEEE/ACM Transactions on*, 2(3):217–225, 1994.
- [13] N. Garg and J. Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.
- [14] A. Jaekel, A. Bari, Y. Chen, and S. Bandyopadhyay. New techniques for efficient traffic grooming in wdm mesh networks. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 303–308. IEEE, 2007.
- [15] B. Jaumard, C. Meyer, and B. Thiongane. Ilp formulations for the routing and wavelength assignment problem: Symmetric systems. In *Handbook of optimization in telecommunications*, pages 637–677. Springer, 2006.
- [16] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the rwa problem. *Discrete Applied Mathematics*, 157(6):1291–1308, 2009.
- [17] G. Jeong and E. Ayanoglu. Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 1, pages 156–163. IEEE, 1996.
- [18] E. Karasan and E. Ayanoglu. Effects of wavelength routing and selection algorithms on wavelength conversion gain in wdm optical networks. *Networking, IEEE/ACM Transactions on*, 6(2):186–196, 1998.
- [19] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 312–320. IEEE, 1982.
- [20] 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 wavelength conversion. *ETRI journal*, 24(2):131–141, 2002.
- [21] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problems. *Journal of Computer and System Sciences*, 50(2):228–243, 1995.
- [22] L. Li and A. K. Somani. Dynamic wavelength routing using congestion and neighborhood information. *IEEE/ACM Transactions on Networking (TON)*, 7(5):779–786, 1999.
- [23] Z. Liu and G. N. Rouskas. A fast path-based ilp formulation for offline rwa in mesh optical networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 2990–2995. IEEE, 2012.
- [24] Z. Liu and G. N. Rouskas. Link selection algorithms for link-based ilps and applications to rwa in mesh networks. In *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*, pages 59–64. IEEE, 2013.
- [25] P. Manohar, D. Manjunath, and R. Shevgaonkar. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *Communications Letters, IEEE*, 6(5):211–213, 2002.
- [26] B. Mukherjee, D. Banerjee, and S. Ramamurthy. Some principles for designing a wide-area wdm optical network. *Networking, IEEE/ACM Transactions on*, 4(5):684–696, 1996.
- [27] A. E. Ozdaglar and D. P. Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking (TON)*, 11(2):259–272, 2003.
- [28] S. Ramamurthy and B. Mukherjee. Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks. In *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, volume 4, pages 2295–2302. IEEE, 1998.
- [29] R. Ramaswami. Multiwavelength lightwave networks for computer communication. *Communications Magazine, IEEE*, 31(2):78–88, 1993.
- [30] R. Ramaswami, K. Sivarajan, and G. Sasaki. *Optical networks: a practical perspective*. Morgan Kaufmann, 2009.



- [31] R. Ramaswami and K. N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking (TON)*, 3(5):489–500, 1995.
- [32] V. S. Shekhawat, D. K. Tyagi, and V. Chaubey. Weight based edge disjoint path routing and wavelength assignment (wedp-rwa) algorithm for wdm networks. In *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*, pages 1–5. IEEE, 2008.
- [33] M. Shiva Kumar and P. Sreenivasa Kumar. Static lightpath establishment in wdm networks new ilp formulations and heuristic algorithms. *Computer Communications*, 25(1):109–114, 2002.
- [34] H. Simonis. Solving the static design routing and wavelength assignment problem. In *Recent Advances in Constraints*, pages 59–75. Springer, 2011.
- [35] H. Siregar, H. Takagi, and Y. Zhang. Efficient routing and wavelength assignment in wavelength-routed optical networks. In *Proc. 7th Asia-Pacific Network Oper. and Mgmt Symposium*, pages 116–127, 2003.
- [36] N. Skorin-Kapov. Routing and wavelength assignment in optical networks using bin packing based algorithms. *European Journal of Operational Research*, 177(2):1167–1179, 2007.
- [37] T. Stern and K. Bala. Multiwavelength optical networks: A layered approach addison-wesley. *Reading, Mass*, 1999.
- [38] S. Subramaniam and R. A. Barry. Wavelength assignment in fixed routing wdm networks. In *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*, volume 1, pages 406–410. IEEE, 1997.
- [39] H. Zang, J. P. Jue, B. Mukherjee, et al. A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks. *Optical Networks Magazine*, 1(1):47–60, 2000.
- [40] H. Zang, J. P. Jue, L. Sahasrabudde, R. Ramamurthy, and B. Mukherjee. Dynamic lightpath establishment in wavelength routed wdm networks. *Communications Magazine, IEEE*, 39(9):100–108, 2001.
- [41] X. Zhang and C. Qiao. Wavelength assignment for dynamic traffic in multi-fiber wdm networks. In *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on*, pages 479–485. IEEE, 1998.
- [42] Z. Zhang and A. S. Acampora. A heuristic wavelength assignment algorithm for multihop wdm networks with wavelength routing and wavelength re-use. *Networking, IEEE/ACM Transactions on*, 3(3):281–288, 1995.