

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:

2021-12-06

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Duarte, I., Cancela, L. & Rebola, J. (2021). Graph coloring heuristics for optical networks planning. In 2021 Telecoms Conference (ConfTELE). Leiria: IEEE.

Further information on publisher's website:

[10.1109/ConfTELE50222.2021.9435591](https://doi.org/10.1109/ConfTELE50222.2021.9435591)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Duarte, I., Cancela, L. & Rebola, J. (2021). Graph coloring heuristics for optical networks planning. In 2021 Telecoms Conference (ConfTELE). Leiria: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/ConfTELE50222.2021.9435591>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Graph Coloring Heuristics for Optical Networks Planning

Inês Duarte², Luís Cancela^{1,2}, João Rebola^{1,2}

¹Optical Communications and Photonics Group, Instituto de Telecomunicações, Lisboa, Portugal

²Department of Science and Information Technology, Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
imlde@iscte-iul.pt; luis.cancela@iscte-iul.pt; joao.rebola@iscte-iul.pt

Abstract—This work focuses on the study of wavelength assignment algorithms based on Graph Coloring techniques. We analyze the performance of the Greedy heuristic, a well-known Graph Coloring heuristic, as well as the Degree of Saturation (DSATUR) and the Recursive Largest First (RLF) heuristics, for planning optical networks. These last two heuristics, to the best of our knowledge, have not yet been applied in the context of optical networks. Extensive simulations have been performed, using real network topologies under a static traffic scenario and we have concluded that the DSATUR and RLF heuristics can outperform the Greedy heuristic in network scenarios where there are several network clusters interconnected by only one or two links. In these cases, the RLF and DSATUR heuristics can provide less 9 and 5 wavelengths, respectively, than the Greedy heuristic, in networks with 34 nodes.

Index Terms—DSATUR, Graph Coloring, Greedy, Optical Networks, RLF, Wavelength Assignment.

I. INTRODUCTION

Optical networks are essential in today's global communications, and the study of planning tools that efficiently allocate network resources is crucial to network providers [1]. Routing and wavelength assignment are two vital functions for allocating the network resources. The routing function consists on finding the best optical path to be used by an optical channel, and implies the selection of a set of links between the source and destination nodes, whereas the wavelength assignment function is responsible for choosing an appropriate wavelength, from the whole wavelength division multiplexing (WDM) signal, for routing the optical channel taking into account the wavelength continuity and the distinct wavelength constraint. These two functions are commonly designated as the Routing and Wavelength Assignment (RWA) problem [1].

RWA can be seen as an optimization problem whose goal is to minimize the number of wavelengths to be used in a static network scenario. For a dynamic network scenario, the goal of RWA is to minimize the wavelength blocking probability, for a fixed number of wavelengths. Moreover, the RWA problem can be solved as a whole or can be solved by finding independent solutions for the routing and also for the wavelength assignment (WA) problems. Usually this last approach is applied to solve RWA problems. Additionally, we can rely, for both problems, on exact solutions, based on Integer Linear Programming (ILP) solutions, or heuristics solutions, that can give approximate or even the exact solutions with shorter computation times [2]. Some examples of the

heuristic algorithms typically used for the routing function are the Dijkstra and Yen's k-shortest path algorithms [3]. Also, the most common algorithms used for the WA problem are the First-Fit and Most-Used. Another type of algorithms that can be used for the WA problem are the Graph Coloring algorithms [4].

Graph Coloring algorithms can be applied in many different areas, besides optical networking, like social networking, chemistry, scheduling, satellite navigation, electrical engineering and computer networking [4]. The problem of Graph Coloring, in short, consists in coloring all the graph vertices with the minimum number of colors so that no vertices connected by an edge are given the same color. In this work, we explore three Graph Coloring heuristics for solving the WA problem in real network topologies assuming a static scenario and a full mesh logical topology. In particular, we study the Greedy algorithm, usually used in WA studies, as well as the Degree of Saturation (DSATUR) and Recursive Largest First (RLF) algorithms [4], [5]. These last two algorithms, to the best of our knowledge, have not yet been applied to WA problems. Besides analyzing and comparing the performance of these Graph Coloring algorithms, using real network topologies, such as COST 239 and CONUS networks, we also compared their performance in terms of the number of wavelengths with the traditional First-Fit and Most Used algorithms. Lastly, we assess and compare the computation time between all the studied algorithms for the various network topologies.

The reminder of the paper is organized as follows. In Section II, the three Graph Coloring algorithms, Greedy, DSATUR and RLF heuristics are explained and their pseudocode are detailed. Also, a validation and a comparative performance study between the algorithms is done in this section. In Section III, the developed planning tool is explained and the real network topologies studied are presented, together with the main features of the network physical and logical topologies. In Section IV, the results are discussed in terms of the estimated number of wavelengths, as well as in terms of computation time required by each one of the algorithms. Finally, the conclusions are drawn in Section V.

II. GRAPH COLORING HEURISTICS

In this section, the pseudocodes of the three Graph Coloring algorithms, Greedy, DSATUR and RLF, are explained. A

comparative performance study between these heuristics is performed and validated.

A. Greedy Algorithm

The Greedy algorithm is perhaps the most used algorithm for coloring graphs and consists in coloring the vertices of the graph one by one according to an arbitrary order [4], for example, placing the vertices in descending order of degree (the degree is the number of incident edges at each vertex). Then, the algorithm assigns each vertex to the first available color. The pseudocode of this algorithm is given in Figure 1.

GREEDY ($S \leftarrow S_1; X \leftarrow V$)

```

(1) for i ← 1 to |X| do
(2)   for j ← 1 to |S|
(3)     if ( $S_j \cup \{v_i\}$ ) is an independent set then
(4)        $S_j \leftarrow S_j \cup \{v_i\}$ 
(5)       break
(6)     else j ← j + 1
(7)     if j > |S| then
(8)        $S_j \leftarrow \{v_i\}$ 
(9)        $S \leftarrow S \cup S_j$ 

```

Fig. 1. Pseudocode of Greedy algorithm.

The pseudocode starts by creating the set X that has all the vertices of the graph in a particular order, e.g., descending. The pseudocode also defines the set S which represents the list of colors associated with the vertices of X. Initially $S = S_1$, which means that S_1 is the first available color to be assigned to a vertex. As the algorithm takes a vertex from the set X, it first tries to assign a color from the set S to that vertex. In each iteration, the algorithm takes each vertex v_i of the set X, and checks if v_i can be assigned to the color S_j , i.e., the algorithm checks if the color S_j was not assigned to an adjacent vertex of v_i (i.e. the vertices associated with color S_j are an independent set of v_i). If the condition is true, then the color S_j is assigned to the vertex v_i and the process moves on to consider the next vertex. Otherwise, if the colors from the set S can not be assigned to that vertex, the algorithm assigns a new color to the vertex v_i . Lines 7, 8 and 9 of the pseudocode represent the creation of a new color and the assignment of that color to vertex v_i , as well as its addition to the set S. The algorithm ends when all vertices of set X have been assigned to a color.

B. DSATUR Algorithm

The DSATUR algorithm is very similar to the Greedy algorithm, since it takes each vertex one by one according to some ordering and then assigns the appropriate color to the vertex. The difference between these two algorithms is on the ordering in which the vertices are colored. In the Greedy algorithm, the vertex ordering is decided before any coloring, whereas, in the DSATUR algorithm, the choice of the next vertex to be colored is decided heuristically based on the characteristics of the current coloring of the graph [4]. In the

DSATUR algorithm, the choice of the next vertex to be colored is based primarily on the saturation degree of the vertices. The degree of saturation of an uncolored vertex is the number of different colors existing in its adjacent vertices [4]. The pseudocode of this algorithm is given in Figure 2.

DSATUR ($S \leftarrow S_1; X \leftarrow V$)

```

(1) for i ← 1 to |X| do
(2)   Choose  $v \in X$ 
(3)   for j ← 1 to |S|
(4)     if ( $S_j \cup \{v\}$ ) is an independent set then
(5)        $S_j \leftarrow S_j \cup \{v\}$ 
(6)       break
(7)     else j ← j + 1
(8)     if j > |S| then
(9)        $S_j \leftarrow \{v\}$ 
(10)       $S \leftarrow S \cup S_j$ 
(11)    $X \leftarrow X - \{v\}$ 

```

Fig. 2. Pseudocode of DSATUR algorithm.

The pseudocode of the DSATUR algorithm (Fig. 2) is very similar to the Greedy algorithm pseudocode (Fig. 1). The main difference between these two algorithms lies in lines 2 and 11 of the pseudocode given in Fig. 2. In each iteration of the algorithm, the next vertex to be colored is selected from the set X (line 2) and a color from the set S is assigned to that vertex, similar to the Greedy algorithm. However, in the DSATUR algorithm, the next vertex to be colored is chosen as the vertex in X that has the maximal saturation degree. If there is more than one vertex with the maximal saturation degree, the one with the highest degree is chosen from these set of vertices. The first vertex chosen to be colored is the vertex with the highest degree. Once the vertex is colored, it is removed from the set X (line 11). The algorithm ends when $X = \emptyset$, which means that all vertices have been assigned to a color of the set S.

C. RLF Algorithm

The RLF algorithm follows a slightly different strategy regarding the Graph Coloring in comparison with the previous algorithms, which have a similar procedure [4]. The RLF algorithm consists in coloring a graph with one color at a time, as opposed to one vertex at a time. At each step, the algorithm applies heuristic methods to identify an independent set of vertices in the graph (i.e. non-adjacent vertices), which are then, associated with the same color. This independent set of vertices is then removed from the graph, and the process is repeated in the resulting smaller subgraphs. The pseudocode of this algorithm is given in Figure 3.

The pseudocode begins by defining four sets. The set X that contains the initially uncolored vertices, the set Z that will aggregate throughout the algorithm the selected vertices in each cycle to be later assigned to a color of the set S, the set S that is responsible for assigning a color S_k to the vertices of Z and the set Y that will contain the uncolored vertices that

cannot be feasible assigned to color S_k . At the beginning of the pseudocode execution $X=V$, $Z=\emptyset$, $S= \emptyset$ and $Y= \emptyset$.

```

RLF ( $S \leftarrow \emptyset$ ;  $X \leftarrow V$ ;  $Y \leftarrow \emptyset$ ;  $Z \leftarrow \emptyset$ )
(1) for  $i \leftarrow 1$  to  $|X|$  do
(2)   for  $j \leftarrow 1$  to  $|X|$  do
(3)     while  $j \neq \emptyset$  do
(4)       Choose  $v \in X$ 
(5)        $Z \leftarrow \{v\}$ 
(6)        $Y \leftarrow Y \cup \Gamma_X\{v\}$ 
(7)        $X \leftarrow X - (Y \cup \{v\})$ 
(8)    $X \leftarrow Y$ 
(9)    $Y \leftarrow \emptyset$ 
(10)  for  $k \leftarrow |S|$  to  $|Z|$  do
(11)     $S_k \leftarrow S_k \cup \{v\}$ 
(12)     $S \leftarrow S \cup S_k$ 
(13)   $k \leftarrow k + 1$ 

```

Fig. 3. Pseudocode of RLF algorithm.

In each outer loop, a color is created. Lines 2-7 are responsible for selecting the vertex to be colored. The pseudocode begins by selecting a vertex v of X (line 4), which is added to the set Z (line 5). All vertices adjacent to v , represented by $\Gamma_X\{v\}$, are then transferred to the set Y (line 6), since they cannot be assigned the same color as v . Finally, on line 7, both the vertex v and its adjacent vertices are removed from the set X , since they are not considered candidates for the assignment of the same color as v . As soon as $X=\emptyset$, no more vertices can be added to the current color. Therefore, in line 8, all the vertices of the set of non-colored vertices Y are moved to the set X , and then in line 9, the set Y is emptied.

In lines 10-12 the assignment of the color selected by the outer loop, S_k , to all vertices that are in set Z and have no color assigned is done. As soon as this color is assigned to these vertices, the color is added to set S . Then, a new color (line 13) is created and the algorithm repeats the loop again. The algorithm ends when both sets X and Y are empty, which means that all vertices have been colored and are in set S .

The process of selecting the next $v \in X$ on line 4, follows a similar rationale to the DSATUR algorithm. The first vertex to be chosen for the assignment of each color is the vertex in X that has the highest degree. The remaining vertices v to be assigned to the same color are selected as the vertices in X that have the highest degree in the subgraph defined by $Y \cup v$.

D. Graph Coloring Heuristics Performance

After the implementation of these algorithms in the software tool developed in this work, the performance of the algorithms was analyzed with random generation of graphs. A random graph, denoted by $G_{n,p}$, is a graph comprising n vertices, where each pair of vertices is adjacent with probability p [4]. In this work, the parameter p of graph $G_{n,p}$ is obtained through the average of the degrees of each vertex, given by,

$$p = \frac{\sum_{i=1}^n \frac{degree_i}{n-1}}{n} \quad (1)$$

where $degree_i$ represents the degree of the vertex i . When $p=0$, it means that all vertices are non-adjacent, and when $p=1$, all vertices of the graph are adjacent.

Figure 4 shows the average number of colors as a function of p obtained for $n=100$, for the Greedy, DSATUR and RLF algorithms. In Figure 4, an upper bound is also represented, which is defined as the highest degree of a vertex to be colored plus one [1]. For each value of p , 50 random graphs are generated. Then, an average is performed to obtain the average number of colors generated by these 50 random graphs realizations. In the Greedy algorithm, the vertices are ordered considering the descending order of degree.

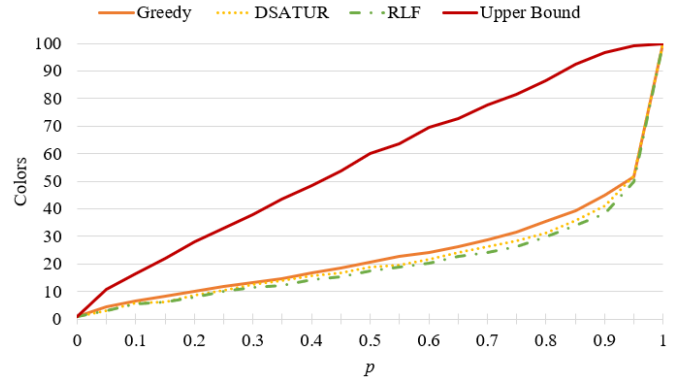


Fig. 4. Average number of colors produced by the Greedy (descending order), DSATUR and RLF algorithms for a network with 100 vertices ($n=100$).

Observing Figure 4, we verify that for values of p close to 1, the number of colors predicted by the different algorithms tends to become similar, reaching the maximum limit of colors i.e., $n=100$ meaning that all vertices are adjacent to each other. We also observe that, the Greedy algorithm is the algorithm that produces the worst results, i.e., it needs to assign more colors, while the RLF algorithm generates the best solutions across the whole set, assigning a lower number of colors. The DSATUR algorithm produces solutions with fewer colors than the Greedy algorithm and slightly more colors than the RLF algorithm. In Figure 4, it is also shown that the results obtained with the upper bound are quite different from the ones obtained with the three Graph Coloring algorithms studied. From the simulations that we have performed for various values of n , we observe that the higher the value of n , the nearer the number of wavelengths estimated by the three Graph Coloring algorithms tends to be.

Analyzing in more detail the solutions produced in Figure 4, for $p \leq 0.2$, the Greedy algorithm achieves about 10.2 colors, the DSATUR algorithm produces approximately 8.7 colors and the RLF algorithm generates 8.04 colors. For $p=0.5$, the Greedy algorithm produces around 20.5 colors, the DSATUR algorithm generates about 18.7 colors and the RLF algorithm generates only 17.5 colors. For this value of p , the upper bound on the number of colors is as high as 60 (almost the triple obtained with the analyzed algorithms), which reveals the poor quality of the considered upper bound. These findings were

also obtained in [4], which allows validating our implementation of the three Graph Coloring algorithms that are going to be used in Section IV for planning real optical networks.

III. DEVELOPED PLANNING TOOL

In this section, we explain the main building blocks of our planning tool with special focus on the implementation of the Graph Coloring algorithms. Then, we present the networks studied in this work, and characterize their physical topologies features and also some of their logical topology features.

The main building blocks of our planning tool are represented in the flowchart of Figure 5. As shown in Figure 5, our planning tool consists on 6 sub-problems:

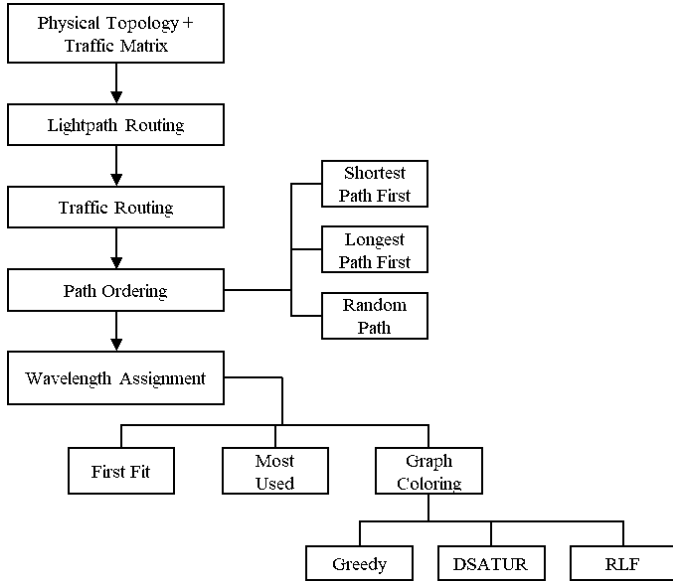


Fig. 5. RWA sub-problems.

- 1) *Physical Topology and Traffic Matrix*: the network physical topology, as well as, the logical topology (characterized by the traffic matrix) are defined.
- 2) *Lightpath Routing*: the lightpaths are routed over a physical topology using the *Yen's k-shortest path* algorithm.
- 3) *Traffic Routing*: after the path is computed and selected, the different traffic units are routed and assigned between the source and destination nodes through the logical topology.
- 4) *Path Ordering*: before assigning the wavelengths to the lightpaths obtained through the routing algorithm, it is necessary to order these lightpaths. The criteria used to order the paths can be shortest path first, longest path first, and random path. Different metrics can be used to establish the paths order, for example, the distance in kilometers or the number of hops.
- 5) *Wavelength Assignment*: after the lightpaths are ordered, the wavelengths are assigned accordingly to a given algorithm. Graph Coloring heuristics such as Greedy, DSATUR and RLF, are going to be used in the RWA planning tool. For these heuristics, step 4 is not required.

Before using the Graph Coloring algorithms, the path graph $G(W,P)$ must be found. This graph is obtained from the graph $G(V,E)$ that represents the physical topology. The vertices of $G(W,P)$ are the optical paths $W = (w_1, w_2, w_3, \dots, w_M)$ and P is the set of links between these vertices [1]. These links are established between one or more vertices (i.e. paths) that share one or more physical links. After obtaining the graph $G(W,P)$, the vertices can be colored considering the three Graph Coloring algorithms studied in this work and explained in Section II. The number of colors obtained corresponds to number of wavelengths needed for solving the RWA problem.

The networks used in this work are: COST 239 [6], [7], NSFNET [8], [9], UBN [10] two variations of the CONUS network [11], [12], one with 30 nodes and other with 60 nodes and two variations of a network generated by the GT-ITM tool in [13], one with 27 nodes and other with 34 nodes.

The GT-ITM (Georgia Tech Internet Topology Modeler) tool generates pseudo-random network topologies on which researchers can perform their analyses [14]. The nodes of the topologies generated are organized in clusters that interconnect with each other by few interconnection links [14]. Figure 6 represents the network with 34 nodes generated by this tool.

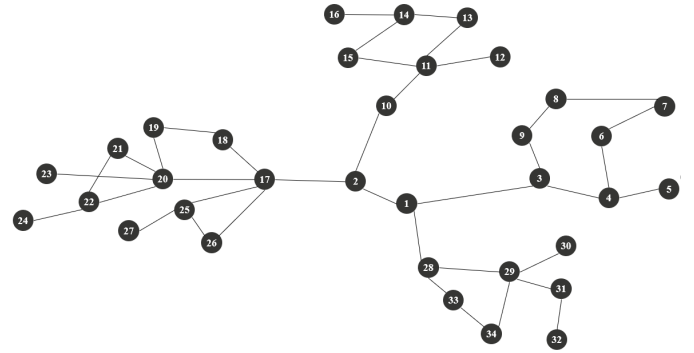


Fig. 6. Physical topology of the network with 34 nodes generated by the GT-ITM tool.

Some of the parameters of the physical topology of the real networks studied in this work are given in Table I.

TABLE I
PHYSICAL TOPOLOGY PARAMETERS.

Network	Node	Link	Average Node Degree	Variance Node Degree
COST 239	11	26	4.7	0.4
NSFNET	14	21	3.0	0.3
UBN	24	43	3.6	0.9
CONUS 30	30	36	2.4	0.4
CONUS 60	60	79	2.6	0.5
GT-ITM 27	27	36	2.7	2.0
GT-ITM 34	34	44	2.6	1.8

The average node degree presented in Table I is defined by [15]:

$$\bar{d} = \frac{\sum_{i=1}^N degree_i}{N} \quad (2)$$

where N is the total number of the nodes of the network. The variance node degree, in Table I, measures the regularity of

the network, i.e., how similar the nodes of the network are in terms of the number of connections, and is defined as [15]:

$$\sigma^2 = \frac{\sum_{i=1}^N (\text{degree}_i - \bar{d})^2}{N - 1} \quad (3)$$

When the variance node degree is zero, it means that all network nodes have the same node degree and the same number of incoming/outcoming connections, as it happens for example in ring networks. As the value of the variance increases, it means that the similarity of the network is reduced.

In Table II, two parameters of the network logical topology used in the context of the Graph Coloring algorithms, the total number of paths of the graph $G(W,P)$ and the parameter p defined in equation (1) for several real networks are presented.

TABLE II
TOTAL NUMBER OF PATHS AND p PARAMETER.

Networks	Total Paths	p
COST 239	55	0.10
NSFNET	91	0.25
UBN	276	0.23
CONUS 30	435	0.38
CONUS 60	1770	0.32
GT-ITM 27	351	0.56
GT-ITM 34	561	0.59

From the analysis of Table II, we observe that the parameter p from the path graph $G(W,P)$ of the networks generated by the GT-ITM tool is $p = 0.56$ with 27 nodes and $p = 0.59$ with 34 nodes. These networks are divided into 3 and 4 domains, with more than 7 nodes inside each domain, which are interconnected with each other by two central nodes. This centralization increases the number of paths that pass through the physical links connected to the central node, increasing the p parameter of the network compared to the remaining real networks. As can be seen in Figure 4, when p is greater than 0.5, the DSATUR and RLF algorithms tend to give even better results than the Greedy algorithm in comparison with $p \leq 0.5$.

IV. RESULTS AND DISCUSSION

In this section, we present the number of wavelengths as well as the computation times obtained by the planning tool developed for the real networks presented in Section III. Besides the three Graph Coloring algorithms we also considered the First Fit and Most Used algorithms. The results presented in this section assume a full mesh logical topology with one traffic unit in each path.

Tables III and IV present the total number of wavelengths obtained by each of the Graph Coloring heuristics and the First Fit and Most Used, for the real networks discussed in Section III. Before applying these algorithms, the network demands are sorted according to three different orderings: shortest path first (SPF), longest path first (LPF) and random path (RP). The values of the random path ordering strategy represent the average obtained after 10 simulation runs.

By analyzing Tables III and IV, we verify that the results obtained by the First Fit algorithm with the LPF ordering strategy, are quite similar to the results achieved by the Graph Coloring heuristics, in particular with the Greedy algorithm with descending order of degree, where exactly the same number of wavelengths is obtained for all studied networks.

TABLE III
NUMBER OF WAVELENGTHS OBTAINED BY FIRST FIT AND MOST USED ALGORITHMS FOR THE NETWORKS DESCRIBED IN SECTION III.

Networks	First Fit			Most Used		
	SPF	LPF	RP	SPF	LPF	RP
COST 239	8	8	8.2	9	8	8.3
NSFNET	24	24	24.0	25	24	24.2
UBN	70	64	66.0	71	64	66.7
CONUS 30	134	123	123.5	135	124	123.8
CONUS 60	550	543	543.5	551	543	543.7
GT-ITM 27	229	221	221.3	229	221	221.1
GT-ITM 34	356	347	347.1	356	347	347.2

TABLE IV
NUMBER OF WAVELENGTHS OBTAINED BY GREEDY, DSATUR AND RLF ALGORITHMS FOR THE NETWORKS DESCRIBED IN SECTION III.

Networks	Greedy	DSATUR	RLF
COST 239	8	8	8
NSFNET	24	24	24
UBN	64	64	64
CONUS 30	123	123	119
CONUS 60	543	543	543
GT-ITM 27	221	217	216
GT-ITM 34	347	342	338

Regarding the behavior of the three Graph Coloring algorithms, we observe that the COST239, NSFNET and UBN networks do not present any differences between the numbers of wavelengths obtained. These results were expected, since the parameter p of each of these networks is less or equal to 0.25, and as shown in Figure 4, the number of colors is quite similar for networks with such p . The same conclusion can be applied to the CONUS 60 network. This network, despite having 1770 nodes, presents a low value of p for the number of nodes it has ($p=0.32$). The CONUS 30 network presents a slight difference in the number of wavelengths obtained by the RLF algorithm (119 wavelengths), since it has a higher p ($p = 0.3792$) and has a smaller number of vertices compared to the CONUS 60 network. Therefore, for the CONUS 30 network, the RLF algorithm gives better results than the other algorithms. The GT-ITM networks, having a p above 0.5, show a clearer difference between the number of wavelengths predicted by the algorithms. For these networks, the RLF algorithm presents the best results reducing the number of assigned wavelengths by 5 for the network with 27 nodes and by 9 for the network with 34 nodes, in comparison with the Greedy algorithm. In this case, the DSATUR also brings advantages, since it assigns less 4 wavelengths than the Greedy algorithm in the network with 27 nodes and less 5 wavelengths in the network with 34 nodes. Therefore, the results for the GT-ITM networks are in agreement with the conclusions taken in

Figure 4, which have shown that, when the p parameter varies between 0.5 and 0.9, the variations between these three Graph Coloring heuristics become more notorious.

Next, the computation time (in seconds) of each one of the wavelength assignment algorithms studied is assessed. The results are shown in Tables V and VI. All the results presented in this work were obtained in a computer with an Intel(TM) core i5 processor at 2.20 GHz and with 8 GB of RAM.

TABLE V
COMPUTATION TIME (IN SECONDS) CONSIDERING FIRST FIT AND MOST USED ALGORITHMS.

Networks	First Fit			Most Used		
	SPF	LPF	RP	SPF	LPF	RP
COST 239	3.9	3.8	3.2	4.0	3.5	3.2
NSFNET	4.7	4.3	3.9	4.9	4.7	3.9
UBN	8.6	7.2	6.5	8.2	7.2	6.5
CONUS 30	8.5	7.2	7.0	8.2	7.4	6.4
CONUS 60	33.3	32.5	31.5	28.9	30.2	30.7
GT-ITM 27	8.5	7.6	6.9	8.8	7.8	6.9
GT-ITM 34	9.2	8.1	7.8	9.6	8.4	8.0

TABLE VI
COMPUTATION TIME (IN SECONDS) CONSIDERING GREEDY, DSATUR AND RLF ALGORITHMS.

Networks	Greedy	DSATUR	RLF
COST 239	4.0	4.2	4.5
NSFNET	4.1	4.7	5.7
UBN	7.2	8.4	9.2
CONUS 30	14.2	16.5	17.5
CONUS 60	162.4	164.4	167.5
GT-ITM 27	8.9	10.7	11.2
GT-ITM 34	16.7	17.4	18.1

The computation time obtained by the First Fit and Most Used algorithms are quite similar to the computation time of the Greedy algorithm for networks up to 25 nodes. For networks above 25 nodes, the First Fit and Most Used algorithms generate solutions quicker than the Greedy algorithm.

Comparing the three Graph Coloring heuristics, the Greedy algorithm, despite producing more colors, generates solutions in a shorter computation time. The RLF algorithm requires more time to generate solutions, but it is the algorithm that assigns fewer colors. The DSATUR algorithm has slightly longer computation time than the Greedy algorithm.

V. CONCLUSIONS

In this work, we have implemented an optical network planning tool that uses three Graph Coloring heuristics to solve the WA problem, the Greedy, the DSATUR and the RLF algorithms.

We have compared the performance of these three Graph Coloring heuristics in some network topologies, COST 239, NSFNET, UBN, CONUS 30, CONUS 60, GT-ITM 27 and GT-ITM 34 in a static network scenario considering a full mesh logical topology with one unit of traffic in each path.

We have concluded that the RLF algorithm produces the best solutions, minimizing the number of wavelengths used

in the network. In addition, for a parameter p above 0.5, it has also been proven that this algorithm and the DSATUR algorithm are advantageous for WA in the planning of optical networks, because they obtain better solutions, for the studied networks, than the most usual wavelength assignment algorithms, the First Fit and Most Used. One such example where these improvements are evident is in networks whose physical topology is based on several cluster of nodes interconnected by central nodes using few links, e.g. as the networks generated by the GT-ITM tool. For this type of networks, more specifically for a GT-ITM network with 34 nodes, the RLF algorithm reduces the number of assigned wavelengths by 9 and the DSATUR algorithm by 5 compared to the Greedy algorithm. The disadvantage of these two Graph Coloring algorithms is slightly higher computation time they require to return a solution, in comparison to First Fit, Most Used and Greedy algorithms.

ACKNOWLEDGMENT

This work was supported under the project of Instituto de Telecomunicações UIDB/EEA/50008/2020.

REFERENCES

- [1] J. Simmons, *Optical Network Design and Planning*, 2nd edition, New York, USA: Springer, 2014.
- [2] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*, 3rd edition, USA: Morgan Kaufmann, 2010.
- [3] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine, SPIE publishers*, vol. 1, pp.47–60, Mar 2000.
- [4] R. M. R. Lewis, *Algorithms and applications in A Guide to Graph Coloring*, Switzerland: Springer, 2016.
- [5] M. Aslan and N. Baykan. A performance comparison of Graph Coloring algorithms. In *International Conference on Advanced Technology Sciences (ICAT'16)*, pp. 266–273, Sept 2016.
- [6] M. Niksirat, S. Mehdi Hashemi, and M. Ghatee, "Branch-and-price algorithm for fuzzy integer programming problems with block angular structure," *Fuzzy Sets Syst.*, vol. 296, pp. 70–96, Aug 2016.
- [7] M. C. Sinclair, Minimum Cost Topology Optimisation of the Cost 239 European Optical Network, in *Artificial Neural Nets and Genetic Algorithms*, pp. 26–29, Springer-Verlag, Vienna, 1995.
- [8] T. L. LaQuey, NSFNET, in *The User's Directory of Computer Networks*, pp. 247–250, Boston: Digital Press, 1990.
- [9] D. L. Mills and H. Braun, The NSFNET Backbone Network, in *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology, SIGCOMM '87*, pp 191–196, New York, NY, USA, 1987.
- [10] E. Biernacka, J. Domzal, and R. Wójcik, "Investigation of dynamic routing and spectrum allocation methods in elastic optical networks," *International Journal of Electronics and Telecommunications*, vol. 63, no 1, pp. 85–92, Feb 2017.
- [11] CONUS WDM network topology, <http://monarchna.com/topology.html>.
- [12] T. Zami, B. Lavigne, and M. Bertolini, "How 64 Gbaud optical carriers maximize the capacity in core elastic WDM networks with fewer transponders per Gb/s," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 1, pp. A20–A32, 2019.
- [13] R. Godoi, P. Hernández, and E. Luque, "Control Evaluation in a LVoD System Based on a Peer-to-Peer Multicast Scheme," *Journal of Computer Science and Technology*, vol. 8, no. 2, pp. 97–103, 2008.
- [14] J. R. Eagan, J. Stasko, and E. Zegura, Interacting with Transit-Stub Network Visualizations, Gvu Center, College of Computing, Georgia Institute of Technology, Atlanta, pp 1–2, 2005.
- [15] C. Fenger, E. Limal, and U. Gliese. Statistical study of the influence of topology on wavelength usage in WDM networks. In *Optical Fiber Communication Conference*, vol. 1, pages 171–173. Baltimore, USA, March 2000.