# MACHINE LEARNING FOR WAVELENGTH ASSIGNMENT AND ROBUSTNESS WITHIN WAVELENGTH-ROUTED OPTICAL NETWORKS

Yashvir Sangha

**3rd Year Project Final Report**

Department of Electronic &
Electrical Engineering
UCL

Supervisor: Polina Bayvel

13 April 2022

I have read and understood UCL's and the Department's statements and guidelines concerning plagiarism.
I declare that all material described in this report is my own work except where explicitly and individually indicated in the text. This includes ideas described in the text, figures, and computer programs.
This report contains 27 pages (excluding this page and the appendices) and 7144 words.

Signed: YSangha  Date: 13/04/22

**Abstract:**

**The routing and wavelength assignment (RWA) problem considers a network where lightpaths can be transported using different optical wavelengths through the network. Existing solutions such as integer linear programming (ILP) are computationally expensive. Machine learning can be used as a computationally inexpensive and accurate solution for finding wavelength requirement and robustness of a network. In this project the foundation is set for implementing an ML model primarily through an efficient data generation pipeline.**

**This project introduces modern fibre optic communication networks and proposes a ML solution for determining the wavelength requirement and robustness of Erdos-Renyi (ER) and Barabasi-Albert (BA) topologies. The following sections describe the theoretical background and technical method for creating a ML regression model.**

**Firstly, a ILP was implemented and tested all unique 3-7 node topologies, with results validated by limiting cut. This method proved to be computationally expensive and difficult to scale for large amounts of data. With 3 nodes taking 0.02 seconds and 7 nodes taking 69 seconds.**

**The machine learning regression model was tested for 11,000 data points containing BA and ER topology data, which resulted in a solution which was 412 times faster than the traditional ILP with an accuracy of 99.7% and mean execution time of 0.48 seconds. The model was trained for 15 node topologies ranging from 50 to 30 edges.**

## Table of Contents

# 1 Introduction and Literature review

## 1.1 Background

In optical networks, light is used to as a means of communication, usually in optical fibres or free space. The fundamentals of optical networks began with Claude Chappe in 1722, transmitting mechanically codes messages over approximately 100km using intermediate relay stations roughly 10 – 15km apart used as the equivalent repeaters [1]. The first testing of the modern optical networks began in 1977 in the UK, and within months, telephone traffic was transmitted at bitrates 45-140Mb/s by BT [2].

Moreover today, over 95% long distance voice and data traffic is carried over optical fibres [3], fast and robust networks form a core technology which drive our current interconnected society. Optical fibres underpin a vast number of current technologies, ranging from data centres to telecommunication systems. As the volume of internet traffic has grown 3.2-fold from 2016 – 2021 [4], with a compound growth rate of 26%, optical networks must improve to demands in terms of, transfer speeds, latency, bandwidth, and robustness.

### 1.1.1 Optical Communication System

A fibre optic system consists of the following elements: optical transmitters, optical amplifiers, optical fibre, and optical receivers. Optical transmitters, typically semiconductor devices, convert an electrical input into a modulated optical signal using various modulation schemes. An optical amplifier is used to negate the transmission loses of the optical signal over distance, this is typically done using an erbium-doped or other fibre amplifiers, covering optical bandwidths beyond approx. 40nm, which amplify multiple wavelengths at high bit rates. The modulated signals are then carried over the optical fibres until reach the optical receiver, where the signal is demodulated and converted back into an electrical signal. This system allows for data transmission which have a higher bandwidth and lower power loss than copper-based or even wireless systems.

Optical transmitters and receivers are combined to create transceivers allowing for bidirectional transmission. Often bidirectional transmission is used to allow communication between both parties. These communication systems can vary in complexity ranging from point-to-point (figure 1) networks to broadcast networks (figure 2) where the signal is transferred to all nodes at the same time.

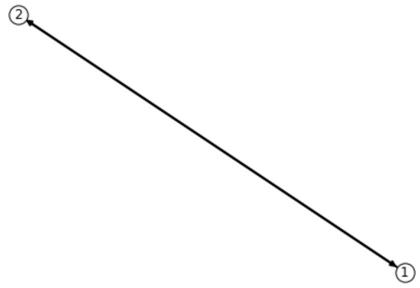


*Figure 1 example of point-to-point network*

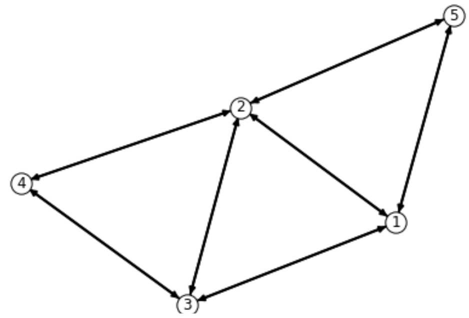


*Figure 2 example of broadcast network*

## 1.1.2 Wavelength routed optical networks

Wavelength-routed optical networks (WRON) refer to communication networks where signals are routed depending on their wavelengths. As seen in figure 1, electrical signals are converted to optical signals using optical transceiver, where they are passed through a multiplexer which divides the bandwidth of the fibre into multiple lower capacity non-interfering wavelengths. These wavelengths are then demultiplexed back into their individual optical signals to be converted back into electrical signals. This process, where data is routed over many wavelengths, is referred to as wavelength division multiplexing (WDM). By having multiple WDM channels across a single fibre, the capacity of a fibre is increased by the number of channels used, allowing for a reduction in the total fibre requirement on a single cable.

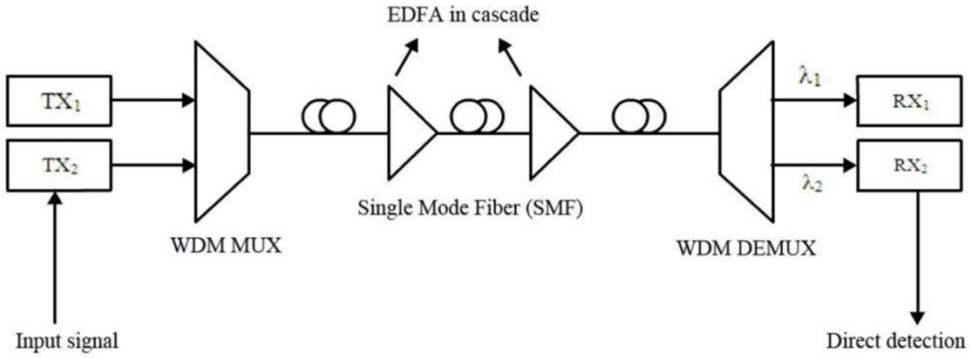


*Figure 3 DWDM (dense wavelength-division multiplexing system)*

However, the number of wavelengths that each fibre can carry is fixed by a given value [5] [6]. Therefore, due to the complex nature of routing and limited wavelength channels, we want to know the requirements of the network are. For example, Wavelength requirement refers to the problem of minimum number of wavelengths required in an optical network, following the wavelength continuity constraints, such that no two paths sharing a link are assigned the same wavelength. Meaning that no two signals on the same path can be assigned the same wavelength. Although, the number of wavelengths available is limited, being much smaller than the total number of paths required, all nodes can typically be routed as wavelengths can be reused as seen in figure 4 which depicts an example optical network topology.

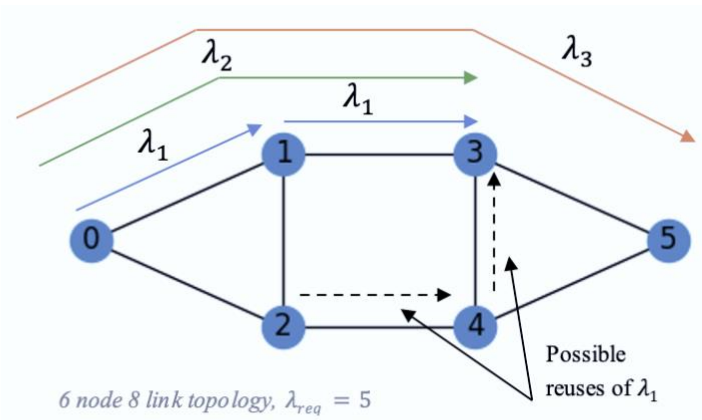


*Figure 4 Wavelength routing and reuse in optical network*

There are multiple benefits to having a system with a lower wavelength requirement. Firstly, from a cost point of view there is higher cost associated with renting fibre from a “dark fibre” provider, as well as the increases cost associated with equipment cost required to “light” the fibre. Secondly, from an operation point of view, networks operators aim to reuse unused wavelengths for better accommodation of future traffic requirements. Finally, the lower the wavelength requirement compared to the capacity of the fibres used will allow for more stable networks with fewer service outages.

As the number of links (edges) within a topology decrease, the number of possible paths from the source to the destination nodes also decreases. This in turn means that there are fewer alternate paths, meaning that more signals will have to be routed along the same edges, therefore increasing the number of wavelengths required upon edge deletion. How resilient the network is to these edge deletions is referred to as the robustness. Therefore, having a lower wavelength requirement which stays low upon edge deletion is indicative of a more robust network.

The exact wavelength requirement of a network topology, for example figure 4, can be determined using integer linear programming (ILP), a mathematical optimisation program discussed further in section 32, however this is only possible for relatively small networks in a reasonable time. As the size of a network increases the computational cost grows exponentially, therefore it is not feasible to use ILP for larger topologies. One avenue to explore is machine learning which offers both fast and accurate estimations of wavelength requirement and robustness.

## 1.2 Literary Review

In a 2010 Bell Labs Technical Journal article by Tkach [7], investigated WDM system trends and extrapolated future requirements, finding that network traffic is increasing at a significantly faster rate than current system capacity growth, noting that there must be an increase in spectral efficiency to meet the demands of the future. This was reaffirmed by Essiambre et. al. 2012 [8], looking specifically into capacity growth within both single and multi-mode fibres and exploration into spatial multiplexing as a solution. These papers highlight the need for more spectrally efficient optical systems. The basis for designing these efficient optical networks is our ability to analyse quickly and accurately model and predict the spectral efficiently of networks. Define spectral efficiency

In the Baroni et. al. 1988 study [9], the authors investigated the wavelength requirement of a small number of arbitrarily connected networks as physical topologies for WRONs and applied a light path allocation algorithm to evaluate the wavelength requirement as a function of physical connectivity, using the results for optimisation and analysis of WRON’s architecture. It was found that increasing the network size led to increased ILP simulation time, due to an increased number of variables and constraints, therefore a faster solution for the wavelength requirement problem must be found.

In 2007, A Jamakovic et al. showed that where was a non-trivial correlation between algebraic connectivity and the robustness of a graph due to node and link failures [10].
The 2009 paper by Châtelain et. al. [11] was able to use topology characteristics, specifically algebraic connectivity, to create formulation to estimate the wavelength requirement for multiple randomly generated network topology with varying nodes with estimation error of up to 42.2% providing over and under estimations for larger networks.

Machine learning is a field of computer science that explores algorithms which can learn and make predictions from data. These algorithms work by creating a model from inputs to create data driven decisions or predictions. Machine learning has evolved from the study of artificial intelligence [12]. An early demonstration of AI was in 1959 [13], where computer program was developed for playing checkers. This was followed by research which aimed to generate human level intelligence through programming large rulesets. This process of defining rules is successful in well-defined systems such as checkers, however, fails in more complex subtle problems such as image recognition and classification and data predication problems. Machine learning is able to solve this problem, through mathematical models based on training data to make predications or decisions.

Machine learning provides a useful solution to the problems faced with optical networks due to its ability to deal with large amounts of complex data and its ability quickly determine and optimise parameters, such as wavelength requirement. For example, in 2019, Martin et. al. [14] used machine learning based routing and wavelength assignment (RWA) to estimate the wavelength requirement of small 5 node topologies, reducing computational time by 93% compared to ILP, however this solution was not scalable and provided the correct wavelength require at a rate of 41% for a larger Abilene network, a legacy ‘Internet2 Network’.

The abovementioned papers highlight the need for solution which is both computationally viable and accurate for the wavelength requirement problem for network capacity and security. This solution can in turn be used to determine the robustness of a topology with respect to wavelength requirement in feasible computational time feasible. Machine learning can meet both the needs of the solution.

# 2 Goals and Objectives

I. Integer Linear Programming

1. Study and gain understanding of integer linear programming and graph theory, specifically regarding wavelength assignment.
2. Research into various ILP formulations and compare against each other.
3. Implement and test ILP for all possible unique topologies ranging from 3 to 7 nodes and verify results.

II. Generating test data

1. Research various graph generation algorithms and create method to generate suitable amount of data and store using database.
2. Create algorithm to delete links from topology data.
3. Explore methods to scale ILP and subsequently use scaled ILP on data.
4. Create metric for robustness and analyse.

III. Machine Learning

1. Research into potential and existing machine learning solutions
2. Research graph topology parameters and calculate said parameters for all topology data
3. Train machine learning model on test data to generate estimates for wavelength requirements and robustness score and validate results
4. Compare accuracy and time of model to ILP

# 3 Theoretical Background

## 3.1 Graph Colouring

The wavelength assignment problem can be described as a graph colouring problem, whereby each optical path is represented through a vertex and if two paths share an optical fibre an edge is formed. If two vertices are connected by an edge, they are considered to be adjacent. Discrete wavelengths are defined as colours in this problem. A colour is assigned to vertex (node) such that no two adjacent vertices share the same colour.

The algorithm for assignment is as follows: [15] [16]

1) Initialize the sets of vertices and edges, $V$ and $E$ respectively.
2) Create vertex $v$ which corresponds to an optical path and add it to the set $V$. Repeat for every added path.
3) If two vertices $v_i$ and $v_j$ share the same optical fibre, connect the vertices with edge $(v, w)$
4) Assign all nodes (vertices) a colour starting with the node of highest degree. If nodes share the same nodal degree assign at random.
5) Move onto the next node of highest degree and check if adjacent node have already been assigned a colour assign a new colour, else assign a colour already in use
6) Repeat sets 4 – 5 until all nodes have been assigned a colour.

## 3.2 Integer Linear Programming

Integer linear programming is used to optimize a linear function which is subject to linear constraints over integer variables. These problems are NP complete (nondeterministic polynomial-time complete) meaning problems which can be solved using brute force algorithms which check every possible solution. These problems are comprised of decision variables (variables which can be controlled), an objective function (a function which needs to be either maximized or minimized) and constraints (conditions the solutions must adhere to).

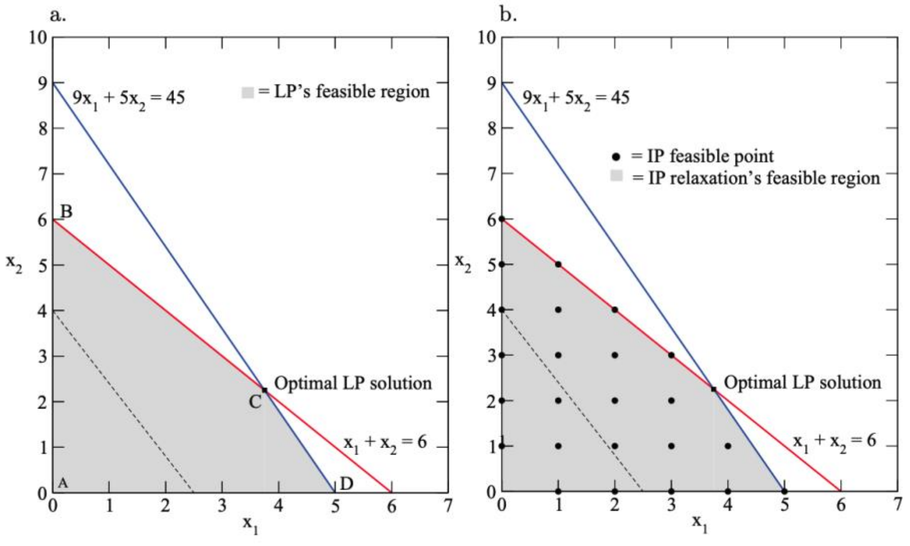


*Figure 5 Graphical representation of linear programming (left) and integer linear programming (right) [28]*

The general form of an integer linear programming minimization problem is as follows:

$$\begin{gathered} Minimise\ f^T x \\ subject\ to\ Ax \le b \\ x \ge 0 \\ and\ x \in Z^n \end{gathered}$$

Where,

$$f^T = [f_1, \dots, f_n] \tag{1}$$

$f^T$ is a vector comprised of known coefficients in Eq. (1).

$$x^T = [x_1, \dots, x_n] \tag{2}$$

$x^T$ is a vector of variables to be calculated in Eq. (2).

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \tag{3}$$

$A$ is the matrix containing the inequality constraints in Eq. (3).

$$b^T = [b_1, \dots, b_n] \tag{4}$$

$b^T$ is the vector of inequality constraints in in Eq. (4).

As opposed to linear programming, every point on the grid within the feasible region must be checked validate the most optimum value of the objective function, whereas within linear programming one of the intersection points must be an optimum solution. This means that as the problem size increases meaning increased number of variables and constraints (in our case number of nodes, links, and max number of wavelengths) the difficulty of the question increases. This in turn requires exponentially increasing computational power, meaning some questions cannot be solved in practical time.

## 3.3 Integer Linear Programming Formulation

The ILP formulation were sourced from [17], which was based on [18]. For a given network, the ILP finds set k of k-shortest paths using Yen's Algorithm [19], which computes a given number of shortest paths between 2 nodes, where k is the number of shortest paths to compute. Yen's algorithm only considers simple paths, meaning nodes are not repeated. Yen's algorithm uses two lists A, which contains the first k-shortest path, and list B which contains potential k shortest paths. The first path is found using any shortest path algorithm (e.g., Dijkstra). Following this the k-1shortest path has each node in its route made unreachable by deleting a given edge and finding the path from the next node to the destination. This creates a new route, which adds a new shortest path. This path is added to list B, given it is not in list A and this is repeated for all nodes in that route, and the shortest path in list B is then added to list A. This is repeated K times. This method is applied to find the wavelength requirement of a system, however, the algorithm has worst case time complexity of $O(N^2)$, where N is the number of nodes in the topology.

Decision variables:

$$\delta_{w,k,z} = \begin{cases} 1\ if\ a\ wavelength\ w\ and\ path\ k\ are\ assigned\ for\ node\ pair\ z \\ 0\ otherwise \end{cases} \tag{5}$$

$$u_w = \begin{cases} 1\ if\ a\ wavelength\ \lambda\ is\ assigned\ at\ least\ once\ in\ \delta_{w,k,z} \\ 0\ otherwise \end{cases} \quad \forall w \in W \tag{6}$$

$$u_w \geq \delta_{w,k,z} \tag{7}$$

$$\forall w \in W, \forall k \in K, \forall z \in Z$$

Let W be the set of wavelengths, K be the set of paths, Z be the set of node pairs (source to destination).

Let $\delta$ be the decision variable such that if a route and wavelength has been assigned then $\delta_{w,k,z}$ = 1, else $\delta_{w,k,z} = 0$ as defined in Eq. (5).

Let $u_w$ be the decision variable such that if wavelength w has been used once $u_w = 1$, else $u_w = 0$, as defined in Eq. (6).

Ensures that each assigned wavelength is used at least once, as defined in Eq. (7).

Constraints:

$$\sum_{w \in W} \sum_{k \in K} \delta_{w,k,z} = 1 \quad \forall z \in Z \tag{9}$$

$$\sum_{z \in Z} \sum_{k \in K} \delta_{w,k,z} I(j \in k) \leq 1 \quad \forall j \in E \quad \forall w \in W \tag{10}$$

Each path may only be assigned one wavelength for all paths z in set Z as seen in Eq. (9).

No two adjacent vertices are to be assigned the same wavelength or assigned more than 1 wavelength ensuring that $\delta_{w,k,z}$ cannot be more than $u_w$ the binary variable which stores if a wavelength and route have been assigned, as seen in Eq. (10).

Objective function:

$$\lambda_n = \min\left(\sum_{\lambda \in W} u_w\right) \tag{11}$$

The objective function of the ILP is set to minimise the total number of wavelengths assigned $\lambda_n$ by minimising the function the sum of $u_w$ , as define in Eq. 11.

## 3.4 Limiting cut

Limiting cut is the lower limit on the wavelength requirement of an optical network [12]. A cut through the topology creates two disjointed self-connected subgraphs which each have $N$ and $N - K$ nodes respectively. As each $\mathrm{K}(N - K)$ node pair requires a route through $C$ links, which are required to crate the two sub graphs, the minimum wavelength assignment required by that given cut $C$ is given by:

$$W_c = \left\lceil \frac{\mathrm{K}(N - K)}{C} \right\rceil \tag{12}$$

Where, $W_c$, rounded to the nearest integer, is the wavelength requirement for a given cut, $K$ is the number of nodes on one side of the cut, $N$ is the total number of nodes in the topology and $C$ is the links intercepting the cut, as seen in Eq. (12).

## 3.5 Graph Models

Graphs models are used to create logical topologies which represent optical networks. Logical topologies refer to the set of optical paths which are established over a physical topology, which refers to the physical layout infrastructure of an optical communication system.

### 3.5.1 Erdos Renyi

The Erdos Renyi (ER) model is the oldest random generative graph model [20]. Given $N$ nodes the probability of edge creation is modeled as independent Bernoulli distributions, where $p$ is the probability of creating an edge between 2 nodes. ER graphs can be characterized by $G(N, p)$, where the number of edges in the graph can be approximated by distribution given by Eq. (13). The generated ER graphs must be biconnected, meaning that if any node is removed the graph will remain connected, to ensure robust optical networks. Typically, ER graphs generates graphs with higher connectivity than is usual for optical network topologies [17].

$$\mathrm{E} = \binom{\mathrm{N}}{2} \mathrm{p} \tag{13}$$

### 3.5.2 Barabasi-Albert

The Barabasi-Albert (BA) model is a generates random scale free network, a network whose degree distribution follows the power law, using a preferential attachment mechanism. Degree, $\delta$,refers to the number of connections each node has to other nodes, therefore degree distribution refers to this probability distribution of degrees across the entire network. The

BA model gives preference to nodes with a higher degree with a higher probability of edge creation. The BA graph model is useful due to the degree distributions being more representative of real graphs [17]. BA graph generation begins with either a seed topology or 2 nodes connected with an edge, where nodes are added sequentially and $m$ edges are connected to every newly added nodes. Edge creation between an existing node, $j$ and newly created node, $i$ is given by Eq. (14). This distribution uses the sum of all the degrees in the current graph and the degree of the new node, $j$ to create large, connected hubs of nodes.

$$p(i,j) = \frac{\delta_j}{\sum_{k \in N} \delta_k} \tag{14}$$

## 3.6 Graph Properties

Within graph theory, graph properties refer characterizations of graphs which are only dependent on abstract structure, meaning the properties of the graph itself not depending on specific drawings or representations of the graph.

### 3.6.1 Connectivity

Connectivity is one of the fundamental concepts of graph theory: it refers to the minimum number of elements which would need to be removed to separate the remaining nodes into two or more isolated subgraphs, which are no longer connected. Both edge and node connectivity were graph properties which were explored when developing the machine learning regression models.

### 3.6.2 Algebraic Connectivity

Algebraic connectivity is a graph property related to the spectrum of corresponding Laplacian matrix. The Laplacian matrix is the degree matrix subtracted from the adjacency matrix, which contain information of the degree of each vertex and whether pairs of vertices are adjacent or not respectively. The set eigenvalues of the Laplacian matrix of a graph are called the Laplacian spectrum [21]. As proposed by Fiedler [22], the algebraic connectivity the second smallest eigenvalue of the Laplacian matrix. Algebraic connectivity is very good indicator of the robustness of a graph, as the larger the algebraic connectivity is the harder it becomes to disconnect the graph into two disjointed graphs [23], as the eigenvalue is only greater than 0 if the graph is connected. Overall, algebraic connectivity is a measure of how well connected the overall graph is.

### 3.6.3 Diameter

Graph diameter refers to the largest number of nodes which must be traversed to travel from one node to another, not considering paths which backtrack, detour or loop. This is done using algorithms to compute all of the shortest paths and returning the maximum shortest path value.

### 3.6.4 Centrality

Within graph theory, centrality is used to identify important element is within a graph. The importance metric depends on the type of centrality metric being investigated, for example, degree centrality ranks nodes depending on its degree whereby the higher the degree of the node the higher the importance.

With respect to wavelength requirement, the max edge betweenness centrality will be used. Edge betweenness refers to the number of shortest paths which go through an edge. An edge with a high betweenness centrality score represents a bridge between two parts of the graph, and therefore, max edge betweenness centrality refers to the maximum betweenness centrality value assigned to an edge. This number represents how critical the highest ranked edge is in the graph, in terms of likelihood of creating two disjointed graphs is removed.

## 3.7 Machine Learning

Within machine learning there are two approaches: supervised and unsupervised learning. Supervised learning is defined by its use of labeled datasets, meaning datasets which already have the correct answers from input data. Supervised learning aims to train a machine learning model with labeled data to be able to make accurate predictions on unlabeled data. Unsupervised learning uses unlabeled data and aims to find hidden patterns in the data.

With supervised learning, the model becomes at risk of over or underfitting. If the model becomes too complicated it risks learning from the noise within the data, leading to the model being unable to generalise for new unlabeled data. However, if the model is too simple it risks not learning the patterns and features of the data, once again making it unable to generalise for new unlabeled data.

## 3.8 Testing and Validation

Underfitting is much less common than overfitting as it is easy to identify by looking indicators during the training stage such as high bias and low variance. Overfitting can quantitively evaluated using cross validation methods, where the data is split into two partitions of training, the data used for training the mode and test data, which is used to validate the accuracy of the training model.

### 3.8.1 Hold out method

The hold out method is one of the simplest and most widely used model evaluation methods. In the hold out method the entire data set is divided into 2 subsets, typically using a 70-30 testing validation split. However, in this method the split occurs randomly, therefore this method leads to highly variable test error rates and high bias as only certain parts of the data train the model.

### 3.8.2 Leave one out cross validation

The leave one out cross validation method divides the data into train and test sets, where one observation is made the test data and all the other data is used for training. This is done n times for all observations and an average of all iterations is calculated and estimated as the test set error. However, this method has very high variance as outputs are likely correlated with each other.

### 3.8.3 K – Folds cross validation

The k folds method splits the data into k sets of equal sizes, the first set is selected as the test set and the k-1 sets are used for training. This process is repeated for all k set, with each iteration calculating the error. The mean of all the error iterations is calculated as the CV test error estimate.

# 4 Technical Method

In this chapter, the approach to the implementation of the integer linear programming, data generation and analysis methods, and the implementation of the machine learning model are described. Firstly, the data generation for the preliminary ILP testing are described. Secondly, the implementation and verification of the ILP is described. Thirdly, the ER and BA data generation and justification of choice of parameters is provided. Finally, the implementation and utilisation of the ML model and subsequent testing and validation is explained.

The entirety of this project was coded in Python, due to easy implementation of automation, simple syntax, versatility brought about through third party libraries, which were used for everything from graph plotting to statistical testing.

Gorobi optimization was used for ILP implementation due to its ease of use, speed compared to its competitors, and community forum support.

A MongoDB database was introduced due to the scale of, and complexity of the data generated. MongoDB also provides the ability the quickly query and add additional parameters to data. MongoDB was instrumental to the workflow of this project, acting as a version control for the data used in the project.

The Networkx library was used for graph data generation and manipulation, due to the having already implemented many of the functions required by the project, for example graph generation functions, edge removal functions, and parameter calculation functions.

The machine learning was done using Skikit-learn, a library which implements machine learning, testing and verification and data analysis methods. Skikit-learn was used alongside the standard data analysis libraries: numpy, matplotlib and statsmodel which were used for mathematical methods, creating graphs and statistical testing respectively.

Throughout the project, a python library called Ray was used, as very early on in the project the sheer computation requirement of ILP was noticed both in terms of time and computational power. Therefore, Ray was used to distribute the workload onto the UCL EEE Budapest cluster CPUs. The UCL Budapest cluster boasts 2 x Intel® Xeon® CPU E5-2690 V4 @ 2.60GHz with 256gb RAM for reference.

## 4.1 Generating preliminary data for ILP

To generate all possible graphs for nodes ranging from 3 to 7, including isomorphic, firstly, we write a script which constructs all possible combinations in lexicographical order. Providing and output as follow (e.g., 3 nodes):

[(0, 1), (0, 2), (1, 2)]
[(0, 1), (0, 2)]
[(0, 1), (1, 2)]
[(0, 1)]

We will filter these results to remove disconnected graphs, permutations, graphs with the incorrect number of nodes and isomorphic graphs. Then the results will be used by network to generate topologies. This was repeated for 3 to 7 nodes. At this point in the project, I was not using Ray or the UCL cluster, therefore preliminary data only went up to 7 nodes as 8 nodes will take hours to generate, and 9 nodes will take days to weeks. In total, 994 unique graphs were generated.

## 4.2 ILP Implementation and Verification

The ILP was developed using the formulations shown in 3.3. Gurobi was used in the functions provided by Robin Matzner's NetworkToolKit library package. The implementation of the ILP formulations was primarily done through multiple nested for loops for the constraints and the configuration of the Gurobi solver and use of the Gurobi ILP function. ILP will be used on the generated 3 -7 node topologies and will return the wavelength requirement, optimization status and time taken. The ILP was set to keep running unto an optimal solution has been found. These results will be compared using the 'cut_size' Networkx is used to calculate the limiting cut wavelength requirement for every preliminary topology.

## 4.3 Baseline ER and BA graphs

The ER and BA generation scripts were developed to:

1. Create and insert the initial topologies with $m = 3$ and $p = 0.45$ with 15 nodes to produce 50 edge topologies into the database, which is 1 fully connected topology.

2. Randomly remove 10 edges in increments of 2 and saving each subsequent topology, ensuring that edge removal does not lead to 2 disjointed graphs and that the minimum degree of the graph is greater than or equal to 2

3. Save each graph and its subsequent graphs into the MongoDB database, with parameters: number of nodes, number of edges, number of edges removed and $m$ or $p$ value.

500 base BA and ER graphs were generated creating a total of 11,000 graphs.

The decision made to not remove any edges which would cause disconnections as this would lead to two separate subgraphs which would have their own individual wavelength requirements.

Note as the ER model is based on probability, a while loop is used to specifically generate 50 edge topologies.

## 4.4 Graph Parameters Generation

The graph parameters described in 3.6 were generated using inbuilt networx library functions. The database was queried for datapoints not containing the wavelength requirement parameter and the ILP was calculated for every queried datapoint, this also allowed for adding graph parameters retroactively. These parameters were added to the database alongside the ILP calculations, and the calculations were distributed on the UCL Budapest CPUs using 1 CPU per calculation. Doing this sped up calculations from 363 hours (extrapolated) to 16.5 hours. A max time of 3 hours was set on the ILP calculations to account for calculations being bottlenecked.

## 4.5 Machine Learning Model

Now that the training data has been generated the data was exported to a .csv file to be used in a Jupiter script to complete the following tasks:

1. Split the dataset into training and validation data
2. Train model using training data using the validation method which results in the highest $r^2$ value and highest accuracy.
3. Validate trained model using test dataset as well as testing model on unlabeled data for different number of nodes and edges.
4. Access the speed and accuracy of the model.

This was done using the sklearn using the linear_model function and validation functions.

# 5 Results, Analysis and Discussion

In this chapter, the results of the verification of the wavelength assignment, and the machine learning model are reconciled and analysed. 11,000 ER and BA topologies were generated and solved via ILP to development a dataset to be used to train a machine learning regression model to predict wavelength requirement and robustness of a given topology. The machine learning model was evaluated in terms of accuracy and speed compared to ILP.

## 5.1 Verification Of ILP

Figures 2 – 6 show the wavelength assignment of all unique 3 ,4, 5, 6 and 7 node topologies using ILP. All results corresponded to their rounded value of limiting cut for each topology. It should be noted that all marked data points represent wavelength assignments, as they can only be discrete, not continuous values.

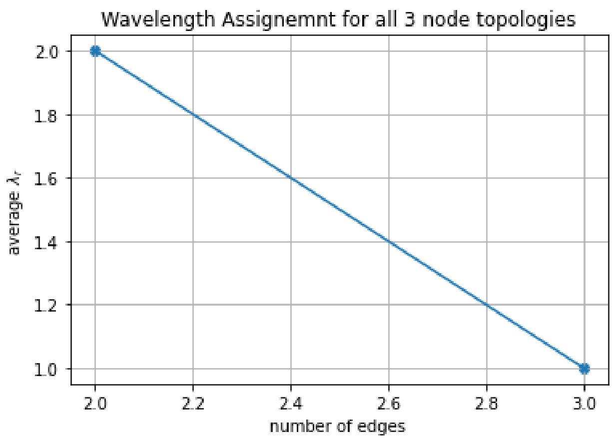


*Figure 6 - Wavelength assignment for 3 node topologies*

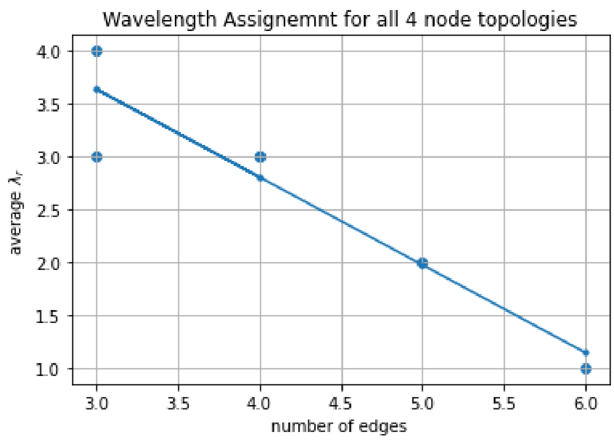


*Figure 7 - Wavelength assignment for 4 node topologies*

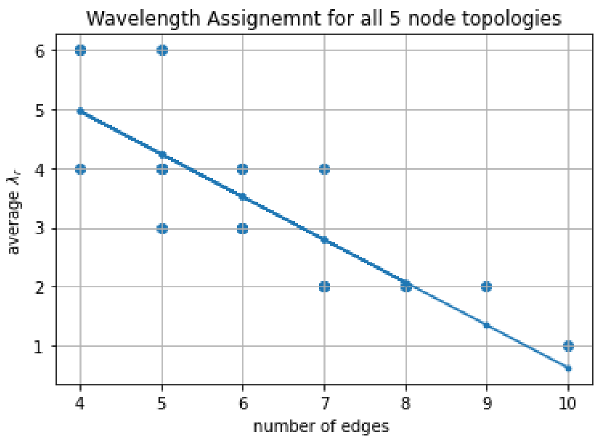


*Figure 9 - Wavelength assignment for 5 node topologies*

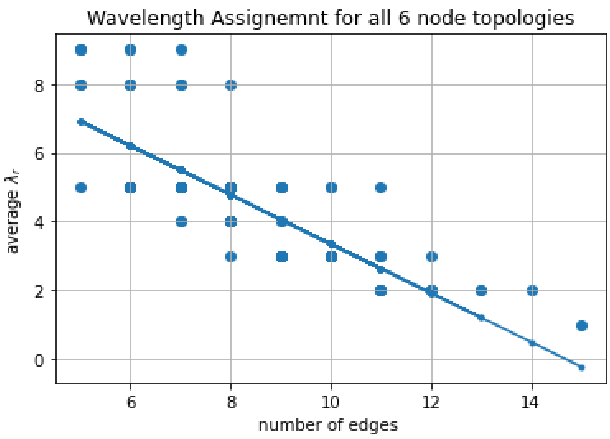


*Figure 8 - Wavelength assignment for 6 node topologies*

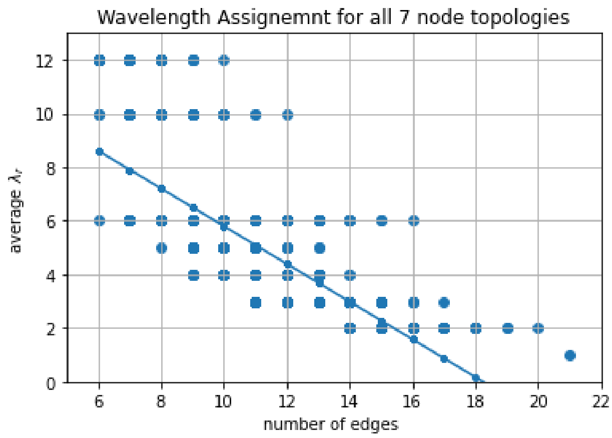


*Figure 10 - Wavelength assignment for 7 node topologies*

Figures 6 - 10, show as expected that as the number of edges increases, and hence connectivity increases, for a graph with the same number of nodes the wavelength requirement also decreases. This is due to there being more links which results in more paths available within the topology, therefore more options for wavelength reuse, allowing for a reduction in network congestion. Whilst the ILP gave correct results, which corresponded to the rounded limiting cuts, using this method for many larger topologies would not be feasible when extrapolating the computational time taken from the verification data. This can be seen through the exponential increase in computational time from 0.02 seconds for 3 node 2 edge topology to 69 seconds for 7 nodes 21 edges.

## 5.2 ILP on Erdos Renyi and Barbosas Albert

To generate training data for the machine learning regression model 11,000 BA and ER graph topologies were generated and ILP was calculated on each one, shown in figure 11, taking 16.5 hours to calculate the ILP for the wavelength requirement.

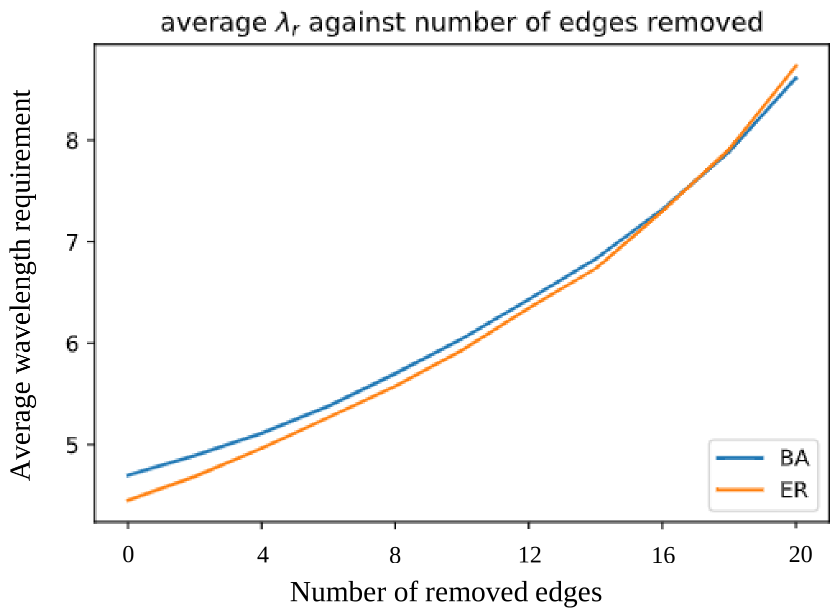


*Figure 11 - Average wavelength assignment of ER and BA topologies*

Figure 11 shows that initially ER graphs have lower wavelength requires up until 17 removed edges where BA graphs have lower wavelength requirements. ER also has a lower robustness throughout due to the larger average gradient. This is likely because BA forms large, connected hubs, which preserve small diameters as edges are removed meaning that there is less disruption to pathing, allowing for a slower increase in the wavelength requirement. Whereas ER graphs are uniformly connected, due to their random edge creation probability causing diameter to increase and edge connections spanning smaller parts of the graph.

## 5.3 Parameter analysis

The graph parameters used for the labelled data for the machine learning regression model must have a statistically significant correlation with wavelength. Therefore, as discussed in the methodology, statsmodel was used to test all the chosen graph parameters. The results of the statistical testing in figure 12.

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| algebraic connectivity | -1.2074 | 0.032 | -37.603 | 0.000 | -1.270 | -1.144 |
| connectivity | -0.0002 | 1.81e-05 | -13.479 | 0.000 | -0.000 | -0.000 |
| diameter | 0.0162 | 0.016 | 1.015 | 0.310 | -0.015 | 0.048 |
| edge conn | 0.1461 | 0.017 | 8.440 | 0.000 | 0.112 | 0.180 |
| edges | -0.0256 | 0.002 | -13.479 | 0.000 | -0.029 | -0.022 |
| max edge | 33.4643 | 0.559 | 59.906 | 0.000 | 32.369 | 34.559 |

*Figure 12 - Statsmodel model summary output*

The coefficient represents the number the parameter must increase by for the wavelength requirement to increase by 1. Figure x shows that as algebraic connectivity algebraic connectivity decreases by -1.2 the wavelength increases, showing that as the graph becomes the graph becomes less connected the wavelength increases, this is the same with the connectivity parameter, however to a much more sensitive extent, as connectivity directly relates to likely the graph is to become disjointed. The same can be seen as the number of edges, as the number of edges decreases the wavelength requirement increases, this is directly due to there being fewer pathing options with few edges in a graph.

The opposite can be seen with diameter and edge connectivity and max edge betweenness centrality (max edge), as diameter and edge connectivity increase the wavelength requirement increases. This is true for diameter as it relates the how long the graph spans, meaning that as the diameter increases the graph is less interconnected meaning fewer paths are available therefore, more wavelengths must be used to route signals. Increasing edge connectivity causes an increase in the wavelength, as the minimum number of edges required to disconnect the graph increases the graph becomes more interconnected. Max edge betweenness centrality refers how critical the most critical edge is in the graph in terms of creating two disjointed graphs. Therefore, as criticality of an edge increases this means the overall graph is less interconnected meaning many paths are routed through one edge, and as the graph is less interconnected the wavelength requirement increases.

These parameters were used to create the labelled data for the machine learning regression model.

## 5.4 Machine Learning Model

The machine learning regression model took in the 11,000 labelled data points for 15 node and 50 – 30 edge ER and BA topologies and was trained with an 80 20 train test split. It was then validated with k folds for 10 folds, resulting in a model with correct prediction rates of 87.8% and 99.7% as shown by figure x and x:

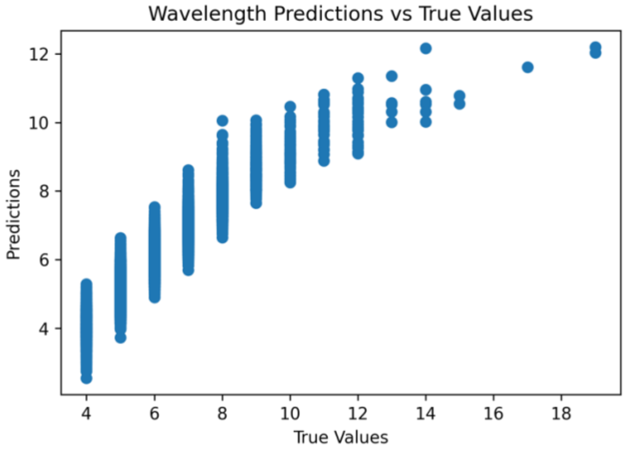

*Figure 14 - Predictions vs True values*

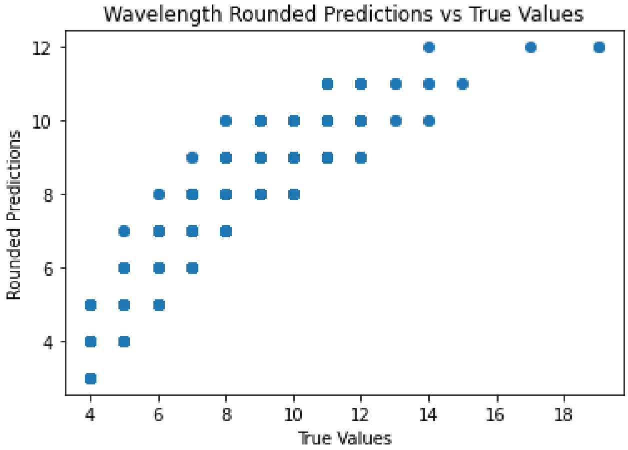

*Figure 13 - Rounded predictions vs True values*

The execution time for the model is 0.0484 seconds, with a standard deviation of 0.001 seconds, meaning the model was very consistent with the time taken to make a prediction. The model has an $r^2$ value of 0.892 meaning the model fit the data quite well, this value can be increase by increasing the number of folds, however this would lead to overfitting of the data. The model had an accuracy of 87.8% before applying rounding to the predictions, which gave an accuracy of 99.7%, however reduced the $r^2$ value to 0.867 when validating with new 11,000 data points for which the solutions were known.

# 6 Conclusion and Future Work

## 6.1 Summary of results and discussion

An ILP was developed which was initially tested on all possible 3 – 7 node topologies and was validated using the limiting cuts method, the ILP was then used to generate 11,000 data 15 node topology with edges varying from 50 to 30 of ER and BA wavelength data alongside various graph parameters. A machine learning model was developed to predict the wavelength requirement and subsequently the robustness of the topology. This model was found to be 412 times faster than the ILP calculations with an accuracy of 99.7% when applying rounding. This improved upon

## 6.2 Future work

This project set the groundwork for the data acquisition and application for a machine learning regression solution. Currently, the data only contain ER and BA data, this could be improved by looking at more types of graph generation methods e.g., Watts-Stogatz or Waxman. Generating more data will be computationally expensive, therefore implementation of heuristics can be used to decrease simulation time. This decreased simulation time can allow for test data generation for topologies with larger number of nodes and edges. Furthermore, additional parameters can be added to the model to increase accuracy, such as nodal variance.

# 7 Bibliography


[1] M. D. Al-Amr, Optics in Our Time, Springer, pp. 178 - 179.

[2] J. Hecht, City of Light: The Story of Fiber Optics, Oxford University Press, 2004, pp. 196-197.

[3] V. Alwayn, Optical Network Design and Implementation, Cisco Press, 2004, pp. 90-91.

[4] A. Sumits, “The History and Future of Internet Traffic,” 28 Aug 2015. [Online]. Available: https://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic.

[5] N. Nagatsu, “Optical Path Accommodation Designs Applicable to Large Scale Networks,” *IEICE Transactions on Information and Systems,* vol. 78, pp. 597 - 607, 1195.

[6] P. Demeester, “Wavelength Routing Algorithms for Transparent Optical Networks,” *European Conference on Optical Communication,* pp. 855-860, 1195.

[7] R. W. Tkach, “Scaling Optical Communications for the Next Decade and Beyond,” *Bell Labs Technical Journal,* vol. 14, no. 4, pp. 3-9, 2010.

[8] R.-J. Essiambre and R. W. Tkach, “Capacity Trends and Limits of Optical Communication Networks,” *Proceedings of the IEEE,* vol. 100, no. 5, pp. 1035-1055, 2012.

[9] P. B. Stefano Baroni, “Wavelength Requirements in Arbitrarily Connected Wavelength-Routed Optical Networks,” *Journal of Lightwave Technology,* vol. 15, no. 2, pp. 242-251, 1997.

[10] A. Jamakovic, “On the relationship between the algebraic connectivity and graph's robustness to node and link failures,” *Next Generation Internet Networks,* pp. 96 -102, 2007.

[11] M. P. B. C. T. F. G. a. D. V. P. Benoît Châtelain, “Topological Wavelength Usage Estimation in Transparent Wide Area Networks,” *J. OPT. COMMUN. NETW,* vol. 1, no. 1, pp. 196-203, 2009.

[12] O. Omer, Introduction to Machine Learning The Wikipedia Guide, Data Science Association, pp. 1-2.

[13] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers,” *IBM Journal of Research and Development,* vol. 3, no. 3, pp. 210-229, 1959.

[14] S. T. J. H. a. A. R. I. Martin, ““Machine Learning-Based Routing and Wavelength Assignment in Software-Defined Optical Networks,” *IEEE Transactions on Network and Service Management,* vol. 16, no. 3, pp. 871-883, 2019.

[15] J. P. J. Hui Zang, “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks,” *Optical Networks Magazine,* 2000.

[16] L. L. B. David W. Matula, “Smallest-last ordering and clustering and graph coloring algorithms,” *Journal of the ACM,* vol. 30, no. 3, pp. 417-427.

[17] R. Matzner, “Making intelligent topology design choices: understanding structural and physical property performance implications in optical networks,” *Journal of Optical Communications and Networking,* vol. 13, no. 8, pp. D53-D67, 2021.

[18] R. J. G. a. P. B. S. Baroni, “On the number of wavelengths in arbitrarily-connected wavelength-routed optical networks,” *Optical Networks and Their Applications,* vol. 15, no. 2, p. MN2, 1997.

[19] J. Y. Yen, “Finding the K Shortest Loopless Paths in a Network,” *Management Science,* vol. 17, no. 11, pp. 661-786, 1971.

[20] P. E. a. A. Renyi, “On the Evolution of Random Graphs,” *Publication of the Mathematical Institute of the Hungarian Academy of Sciences,* pp. 12-61, 1960.

[21] R. M. Robert Grone, “THE LAPLACIAN SPECTRUM OF A GRAPH II*,” *SIAM Journal on Discrete Mathematics* , vol. 7, no. 2, pp. 221-229, 1994.

[22] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak Mathematical Journal,* vol. 23, pp. 298-305, 1973.

[23] A. J. a. S. Uhlig, “On the relationship between the algebraic connectivity and graph's robustness to node and link failures,” *Next Generation Internet Networks,* pp. 96-102, 2007.

[24] Ivan Djordjevic, Coding for Optical Networks, Boston: Springer US, 2010, p. 2.

[25] A. H. Gnauck, “Optical Phase-Shift-Keyed Transmission,” *Journal Of Lightwave Technology,* vol. 23, no. 1, pp. 115 - 130, 2005.

[26] H. ZANG, “A Review of Routing and Wavelength Assignment Approaches for WavelengthRouted Optical WDM Networks,” *Baltzer Science Publishers.*

[27] D. W. Matula, “Smallest-last ordering and clustering and graph coloring algorithms,” *ACM.*

[28] M. L. James Clarke, “Global Inference for Sentence Compression An Integer Linear Programming Approach,” *Journal of Artificial Intelligence Research,* vol. 31, pp. 399-429, 2008.