Available online at www.sciencedirect.com
ScienceDirect
ICT Express 9 (2023) 890–895
www.elsevier.com/locate/icte

Triggering strategy for defragmentation process in Elastic Optical Networks using Machine Learning techniques

Enrique Dávalos[a,*], José-Luis Enciso[a], Nicolás Silva[a], Juan Pinto-Ríos[b], Ariel Leiva[b]

[a] *Polytechnic Faculty, Universidad Nacional de Asunción, Campus Universitario UNA, San Lorenzo, Paraguay*
[b] *School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2950, Valparaíso 2362804, Chile*

Received 8 July 2022; received in revised form 3 January 2023; accepted 24 January 2023
Available online 30 January 2023

Abstract

Bandwidth fragmentation is a critical problem for Elastic Optical Networks (EON), and spectrum defragmentation is the most important strategy to mitigate this phenomenon. In this work we propose a Machine Learning (ML) based method for estimating the Blocking Rate, which, when exceeding a threshold, triggers a defragmentation process. This is done in order to achieve better results in terms of the number of blocking demands and the number of re-routed connections. The performance of the proposed method was compared with two other known strategies: fixed-time (FT) defragmentation, and triggering based on one fragmentation metric (BFR). Simulation results were evaluated using two multi-objective metrics. Experimental results show that the proposed method is more efficient than the other two, being the best method in 85.7% of comparisons using the Pareto Coverage metric, and obtaining 47.4% of non-dominated solutions in the Pareto Front.

Keywords: Elastic Optical Networks; Link fragmentation; Spectrum defragmentation; Machine Learning

1. Introduction

The ever-increasing popularity of the internet and web-based services, such as Content Delivery Networks (CDN) and Video on Demand (VOD), has produced a vast growth of the bit rate demands on computer network carriers. This new reality forces us to study new and ad-hoc technologies related to the transmission of data [1] to deal with the saturation of the capacity of current optical networks, a phenomenon known as "capacity crunch".

The coarse bandwidth granularity of traditional and current optical Wavelength Division Multiplexing (WDM) networks leads to inefficient use of spectrum since each bit rate demand is assigned to a fixed spectrum portion even if it might require lower bandwidth. This disadvantage gives rise to Elastic Optical Networks (EON) [2], which arise as a solution to the aforementioned problem, providing greater flexibility in the division of the spectrum, and, therefore, the requirements can be allocated more efficiently.

Due to dynamic traffic, and continuity and contiguity constraints in transparent or all-optical networks, the phenomenon called *Bandwidth Fragmentation* [3] emerges. It occurs when available Frequency Slots (or FSs) are found separated in the link spectrum forming isolated blocks. These blocks could be unusable for new connections because they are not able to meet the demand due to the restrictions mentioned above, consequently, demand blocking probability [4] increases considerably.

The fragmentation problem of EON networks has been widely studied in the current literature and several strategies have been proposed to mitigate it. The main approach consists in executing a *Spectrum Defragmentation* process [5] periodically. Defragmentation consists of the reconfiguration or re-routing of a sub-set of connections already established within the network in order to reduce spectrum fragmentation by removing non-contiguous free FS blocks.

Considering proactive approaches to solve the problem of fragmentation, we found that spectrum defragmentation processes sometimes execute in periods where they are not entirely necessary. That is, the network is in a state of low fragmentation and a low connection request rejection rate, which causes inefficient defragmentation, a greater number of connection cuts, and an unnecessary increase in processing costs.

Machine Learning (ML) is a branch of Artificial Intelligence based on the idea that computational systems can learn

* Corresponding author.
E-mail address: edavalos@pol.una.py (E. Dávalos).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2023.01.008
2405-9595/

from data and identify patterns with no human intervention. ML techniques were already being used in optical network problems, as seen in [6].

In the following sections, we present a novel ML based estimator for blocking rate which is implemented with Machine Learning techniques. This is used as a trigger for the defragmentation process. It takes the instant value of several fragmentation metrics of the EON as an input for an Artificial Neural Network and estimates a parameter that can be seen as a blocking rate for future time periods. Simulations verified the efficiency of this approach in terms of number of blocked demands and number of reconfigurations when compared to periodic processes and other methods which consider the value of one fragmentation metric.

This work is organized as follows: Section 2 is a brief review of previous works about how to trigger the defragmentation process and different applications of ML techniques to EON-referred problems. Section 3 shows our proposed method to trigger the defragmentation process, while Section 4 presents an application example. Section 5 summarizes conclusions and future work proposals.

2. Previous works

2.1. Triggering of the defragmentation process

There are a few works in the literature that analyze the triggering of the defragmentation process. Authors of [7] present an analysis of periodic defragmentation in EON networks. The approach consists of carrying out the defragmentations in fixed periods, each N time slot, with the main objective of finding optimal values of N. This type of defragmentation in fixed periods is widely used in research such as [8,9], among others. Our work uses this technique in order to compare the results with our method.

Authors of [10] propose a mechanism to trigger the defragmentation process based on the value of the *High-slot Mark* (HM) which indicates the maximum value of an occupied slot in the network. The defragmentation process is triggered randomly when the value of HM is greater than the value of HM_{max} which was previously defined. For the triggering process in our proposed method, a set of features that indicate the current state of the network is used, in which the High-slot Mark is included, also called *Maximum Slot Index* (MSI).

Authors of [11] combine the reactive and the proactive approach to determine the period in which the defragmentation process will be executed. For proactive defragmentation, the proposed method uses the number of released connections as a trigger, so it can be deemed periodic.

Finally, the authors of [12] present a complete analysis of the EON network defragmentation problem. To determine when to reconfigure, they propose a triggering algorithm that considers the instantaneous blocking rate in a previous period of time and the bandwidth utilization. They trigger defragmentation when these indicators overpass certain values, and when the network encounters a growth in bandwidth utilization. Instead, our method seeks to avoid exceeding this blocking rate threshold by triggering the defragmentation process before this happens.

In the following sections, we present an ML-based estimator of intelligent triggering, which takes into account numerous factors such as metrics of network fragmentation, network utilization, and demand blocking.

2.2. Machine learning in EON networks

Machine Learning is a branch of artificial intelligence that builds a mathematical ML-based estimator from sample data, known as training data. In the context of EONs, this has been defined through three paradigms [6]: Predictions, decision-making, and regressions. This allows the collected data to be used to perform these tasks without having been explicitly programmed to perform such a task before. For example, in the predictions paradigm, the evaluation and forecast of QoS [13] can be done with a well-planned data collection. In the decision-making paradigm [14], the authors proposed a Deep Reinforcement Learning (DRL), to develope an autonomous ML-based estimator of Routing, Modulation, and Spectrum Assignment (RMSA) in EON and multi-band EON, using dense neural networks to process the feature's extraction of the EONs they proposed. Therefore, the DRL agent can interact with the environment and learn the policy to allocate the optical resource. In the regression paradigm, in [15] real traffic data is used in different topologies, relating them through an ML-based estimator using a logarithmic regression as a basis to approximate common patterns for routing.

In the area of interest of this work, we find some research that solves the problem of fragmentation but focuses on another type of network, such as the one presented in [16], which is focused on *Space Division Multiplexing Elastic Optical Networks* (SDM-EON). They used *Elman neural networks* for predicting traffic in order to mitigate fragmentation and the problem of cross-talk.

In [17] the authors propose a defragmentation algorithm training using an unsupervised learning approach, so it does not require prior knowledge of the network. The algorithm is responsible for identifying those lightpaths that can be grouped based on a certain feature, in order to later map those groups and reorder the spectrum without the need to perform re-routes.

To the best of our knowledge, no previous work has been done that applies ML techniques to seek the best moment to begin a defragmentation process, which is the objective of our work.

3. Proposed methodology to create the ML-based estimator of blocking rate for EON

In this work, an estimator based on an artificial neural network (ANN) is proposed to perform a regression to provide a value of the blocking rate BR_t, given a set of the EON's features. A defragmentation process is triggered if the estimator indicates that this parameter exceeds a certain threshold. With this, we avoid the network performance from being affected.

To create the estimator, a 3 stage methodology is proposed. These are: `Stage 1`: A database containing fragmentation metrics and bandwidth utilization (detailed in Section 3.1) obtained via simulation must be created, which must consider several network topologies at different traffic loads, measuring the blocking rate given a certain period of time; `Stage 2`: A fraction of the created database (training database) must be used to train a neural network in order to periodically predict the blocking rate, given a set of features of the EON's at that moment as an input of the estimator; `Stage 3`: The ML-based estimator must be validated with a database not used through the training of the estimator. The validation database must be used to perform hyperparameter adjustments to the ML-based estimator. After performing the training validation, the ML-based estimator is ready to be tested. The sections below provide more details about the different stages.

3.1. Stage 1: Database creation via simulation

To get the data for training the neural network, a discrete event simulator that considers the features and metrics indicated in the context of a dynamic EON operation for different topologies must be used. The generation of connection requirements and releases must be performed with different traffic loads and patterns as a function of time, like tidal traffic modeling for suburban, commercial, and cloud internet traffic [18].

The fragmentation metrics that must be used as features are the following: (i) Utilization entropy (UE) [19], (ii) Shannon entropy (SHF) [20], (iii) Bandwidth Fragmentation Ratio (BFR) [21], (iv) Maximum Slot Index (MSI), and (v) Spectrum Consecutivity (EC) [22].

An applied feature related to the network bandwidth utilization metric is Network Utilization (NE): defined as the quotient between the sum of all the occupied FSs at a certain moment, and the total number of FSs in the network. A feature related to the instant blocking metric is the Total of blocked FSs (BFS): value that shows the sum of the FSs required by blocked demands in a D demands size of a window of already processed demands prior to the current period of time.

Data pre-processing must be considered in order to clean the data and calculate the blocking rate, considering a time window ΔT.

The created database must be split into two batches, the first one for training (Stage 2), and the second one for validation (Stage 3).

3.2. Stage 2: Creation and training of the ML-based estimator

An ANN architecture for the ML-based estimator must be defined considering seven inputs, which are the features extracted from the training data set. An initial combination of a number of hidden layers and neurons for each layer must be defined, to be tuned during the validation process (Stage 3). For the output of the ANN, an `activation function` must be used, which must allow to return the value of the regression estimating the value of BR_t. It is defined as the rate of blocking in D future demands. For the training process, as the value of the blocking rate to estimate is already known (output of `Stage 1`), it is necessary to calculate a metric to relate the difference between the estimated value of the ML-based estimator and the real value of the BR_t in the training database. For this coarse fit of the estimator, a sufficient number of epochs must be considered, until the value of the training error metrics stops improving. The weights of the ANN obtained during the training process must be saved. These parameters are the ones that will define the initial estimator adjustments in stage 3.

3.3. Stage 3: Validation of ML-based estimator

To verify the effectiveness of the estimator it is necessary to use the validation database (which was not included in the training) to make predictions of the blocking rate. To ensure greater accuracy in the regression task, a hyperparameter adjustment process using a neural architecture search (NAS) method for the ML-based estimator must be done, considering training error metrics using the same method of Stage 2 that calculates the error between the estimate done by the model and the real value in the validation database.

With these two parameters, the ANN's hyperparameters must be defined to get better regressions with the best ML-based estimator according to these last values. For this adjustment, a grid search is a great guide to define the hyperparameters [23].

4. Application example

4.1. Creation, training and validation of the ML-based estimator

Considering the methodology proposed in Section 3, an ML-based estimator was designed, starting with a database creation which was divided according to parameters shown in Table 1. This considered a total of 120,000 demands per simulation. For the creation and training process (Stage 2), an initial model of 7 inputs and a single layer of 16 neurons was considered. The training error metrics used are mean absolute error (MAE) and mean square error (MSE). The considered number of epochs was 10.000 before the training stops. For the validation process (Stage 3), the NAS method used was the Grid Search [26]. The final architecture after the hyperparameters tuning is an ANN with an input layer with 7 inputs, two fully connected layers of 64 and 32 neurons each, considering all the layers using a ReLu [26] activation function. This model obtained an MAE and MSE equal to, 0.0239 and 0.002, respectively.

4.2. Using the ML-based estimator in the defragmentation triggering process

To prove that the ML-based estimator proposed can improve the defragmentation trigger process, it has been implemented in a simulator [8] (considering 40,000 connection

Table 1
List of parameters considered for the ML-Based estimator creation process.

Stage 1	Database creation
Discrete event simulator	EON simulator developed in [8]
Resource allocation algorithm	Fragmentation-aware Routing and Spectrum allocation (FA-RSA) [24]
Fragmentation Metrics	UE, SHF, BFR, MSI, EC
Bandwidth utilization metrics	NE
Instant blocking metric	BFS
Topologies	USNet, NSFNet
Traffic pattern	Tidal traffic modeling: Suburban, Commercial and Cloud [18]
Traffic load	100 to 700 Erlangs
FSs per link	320 FSs
FSs slot required by demand	Randomly between 1 and 8 FSs
D-demand window size	30
Pre-processing ΔT	10
Database split	70% for train, 30% for validation [25]
Stage 2:	**Training the ML-based estimator**
ANN inputs	Metrics of Stage 1
ANN Output	Blocking rate estimation
Activation function	Regression Task domain
Epochs	Until Training error metrics stops improving.
Stage 3	**Validation**
Training error metrics	MAE, MSE.
NAS	Grid Search [23]

requests) to estimate a value of BR_t in a future number of demands D in the NFSNet topology with different traffic patterns (suburban, commercial, and cloud). This traffic is different from the one used in Stage 2. If this estimated value exceeds a threshold value BR_{th}, the defragmentation process is triggered. For the defragmentation process, the genetic algorithm proposed in [8] was applied.

The proposed method (PM) is compared with two other known methods for triggering: (i) Fixed-time (FT) defragmentation, which consists of defragmenting the network every certain constant number of units of time. (ii) BFR defragmentation, which consists of defragmenting when the BFR fragmentation metric reaches a specific threshold value BFR_{th}.

Each method has its own parameter for triggering the process of defragmentation and during the tests, this parameter is changed three times. Then each scenario was repeated five times. This produces the same number of solutions for each method. The parameter values were chosen by taking as a base the number of defragmentations that the fixed-time method performs, and looking for a way in which they perform similar amounts of defragmentations. In this way, we make the comparison as fair as possible. In this example of application, we consider the threshold values shown in Table 2.

Table 2
Threshold values for defragmentation triggering.

Defragmentation method	Threshold value
PM	BR_{th}: 0.39, 0.40, 0.41
FT	Times units: 100, 150, 200
BFR	BFR_{th}: 0.81, 0.82, 0.83

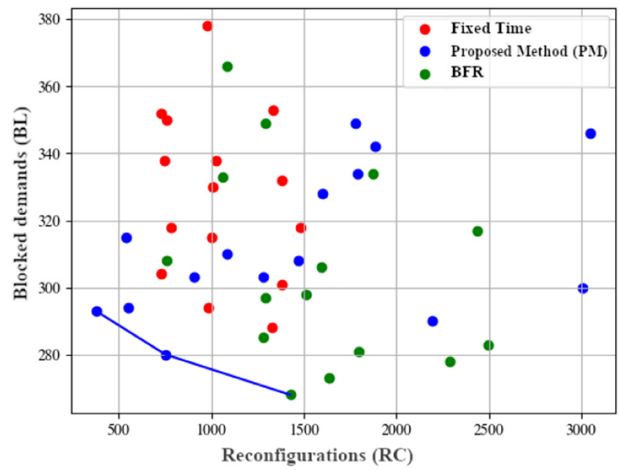

Fig. 1. Solutions for NSFNET topology considering a suburban internet pattern traffic [18].

4.2.1. Objective functions to analyze

For the evaluation of the results, we considered two global *objective functions*, measured at the end of each simulation: (i) Number of blocked demands (BL): The sum of the blocked requests during the simulation. (ii) Number of reconfigurations (RC): Number of reconfigured connections during the defragmentation process carried out in the simulation. In this context, a *solution* is a pair of values for the two objective functions, the result of the simulation execution using this method, and a determined triggering parameter. The goal is to achieve solutions that have fewer blocked requests and connection reconfigurations. However, the improvement of one objective function could mean the worsening of the other.

Two multi-objective metrics were considered: (i) Number of solutions in the combined Pareto Front (SPF): It indicates the number of solutions that the considered method contributes to the combined Pareto Front (formed with non-dominated solutions of all methods). The quotient of this number with the total number of solutions in the combined Pareto Front is also known as the Contribution metric [27].(ii) Pareto Coverage (PC) [28]: This performs the comparison of the Pareto front solutions of the proposed method with the Pareto front solutions of the other methods, taking them in pairs.

4.2.2. Results analysis

In Figs. 1 to 3 we can observe the results for the three different types of traffic patterns applied to NSFNet topology. The Pareto Front is shown as a blue line. A predominance of the proposed method can be seen, in terms of the number of solutions, in two of the three traffic variation types.

The experiments carried out, show that all of the experimental instances found in the combined Pareto Front get the following values: $PM = 9$, $FT = 4$ and $BFR = 6$. The proposed method (PM) in this work represents 47.4% of the points within it, which indicates that this strategy produces more non-dominated (and better) solutions. It is an indicator of greater efficiency in terms of minimizing the number of blockages and the number of reconfigurations in the network.

Table 3
Pareto Coverage for NSFNET topology.

NSFNET					
Traffic	A	B	C(A,B)	C(B,A)	Conclusion
Suburban	PM	FT	0.25	0	A covers B by 25% and is covered by 0%
Suburban	PM	BFR	0.43	0	A covers B by 43% and is covered by 0%
Commercial	PM	FT	0	0.29	B covers A by 29% and is covered by 0%
Commercial	PM	BFR	0.33	0.43	A covers B by 33% and is covered by 43%
Cloud	PM	FT	1	0	A covers B by 100% and is covered by 0%
Cloud	PM	BFR	0.67	0	A covers B by 67% and is covered by 0%

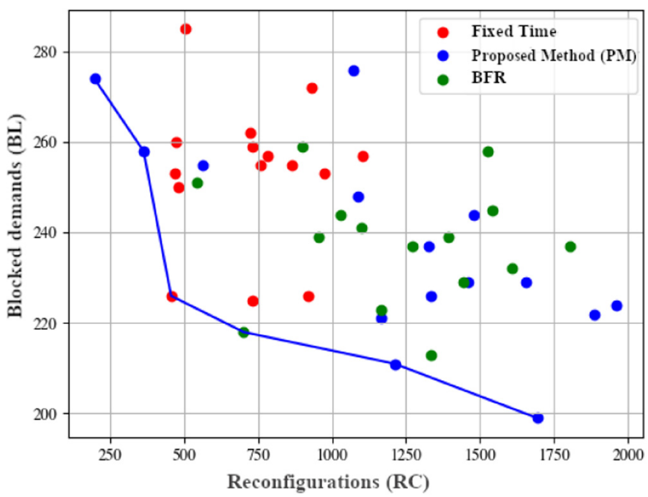

Fig. 2. Solutions for NSFNET topology considering a commercial internet pattern traffic [18].

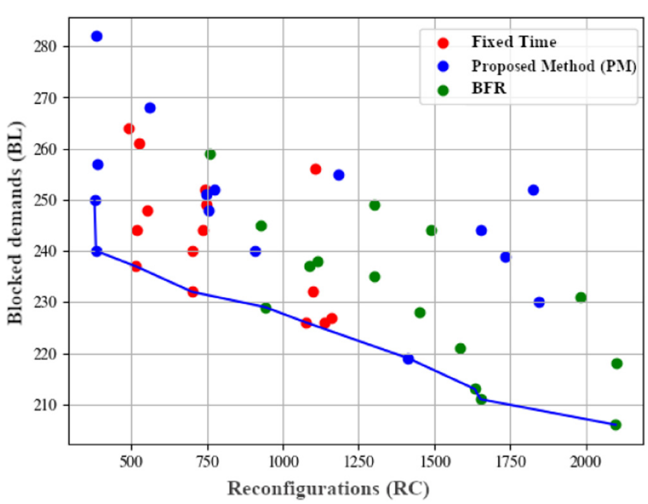

Fig. 3. Solutions for NSFNET topology considering a cloud internet pattern traffic [18].

Considering the BL values shown in Figs. 1 to 3, the network blocking probability is around to 10^{-2}.

Table 3 presents the results of the PC metric [28]. In 85.7% of all comparisons, the proposed method obtains greater coverage in the optimal solutions in the NSFNet topology.

5. Conclusions

A Machine Learning (ML)-based method for estimating the blocking rate (BR_t) in a dynamic EON was proposed. This estimator is used to trigger a defragmentation process, which when exceeding a threshold value of BR_{th}, triggers a defragmentation process to reach better results in terms of the number of blocking demands and the number of re-routed connections.

To evaluate the efficiency of the proposed triggering method, three different scenarios were considered with a variable volume of traffic and using the NSFnet topology. The objective functions to be optimized were: (i) the number of blockings for a certain test instance, and (ii) the number of reconfigurations, done by the defragmentation processes, at the end of each test instance.

Experimental tests were carried out to compare our method of triggering against two others taken from the literature: (i) Fixed-time defragmentations, which is a widely used strategy, and (ii) triggering defragmentations considering the current value of a single fragmentation metric (BFR).

To compare the results obtained concerning the aforementioned objective functions, two performance metrics were used for multi-objective optimization: (i) Number of solutions in the Pareto Front (SPF) and (ii) Pareto Coverage (PC). As a result of comparing the methods based on the previously stated objectives, it is concluded that the proposed method is better since it achieves better results in most scenarios, minimizing the values obtained for BL and RC. Considering the SPF metric, it is obtained that it constitutes 47.4% of non-dominated solutions and in the case of PC, it achieved favorable results to the proposed method in 85.7% of the total comparisons made.

Further research is recommended in order to carry out a greater number of experiments, considering different topologies, physical phenomena, and neural network architectures, among others. This will allow us to generate general recommendations.

CRediT authorship contribution statement

Enrique Dávalos: Conceptualization, Formal analysis, Supervision, Writing – original draft. **José-Luis Enciso:** Investigation, Software, Data curation, Visualization. **Nicolás Silva:** Software, Data curation, Visualization. **Juan Pinto-Ríos:** Methodology, Validation, Resources. **Ariel Leiva:** Writing, Review and editing, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support from projects: ANID Doctorado Nacional (2022-21220867), is gratefully acknowledged.

References

[1] M. Aibin, K. Walkowiak, Defragmentation algorithm for joint dynamic and static routing problems in elastic optical networks with unicast and anycast traffic, in: 2016 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2016, pp. 1–5.

[2] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, S. Matsuoka, Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies, IEEE Commun. Mag. 47 (11) (2009) 66–73.

[3] B.C. Chatterjee, N. Sarma, E. Oki, Routing and spectrum allocation in elastic optical networks: A tutorial, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1776–1800.

[4] W. Shi, Z. Zhu, M. Zhang, N. Ansari, On the effect of bandwidth fragmentation on blocking probability in elastic optical networks, IEEE Trans. Commun. 61 (7) (2013) 2970–2978.

[5] B.C. Chatterjee, S. Ba, E. Oki, Fragmentation problems and management approaches in elastic optical networks: A survey, IEEE Commun. Surv. Tutor. 20 (1) (2017) 183–210.

[6] R. Gu, Z. Yang, Y. Ji, Machine learning for intelligent optical networks: A comprehensive survey, J. Netw. Comput. Appl. 157 (2020) 102576.

[7] J. Comellas, L. Vicario, G. Junyent, Periodic defragmentation in elastic optical networks, in: 2018 20th International Conference on Transparent Optical Networks, ICTON, IEEE, 2018, pp. 1–4.

[8] E.J. Dávalos, M.F. Romero, S.M. Galeano, D.A. Báez, A. Leiva, B. Baran, Spectrum defragmentation in elastic optical networks: Two approaches with metaheuristics, IEEE Access 7 (2019) 119835–119843.

[9] J. Luo, Z. Zhang, W. Sun, W. Hu, Partial defragmentation in flexible grid optical networks, in: Asia Communications and Photonics Conference, Optical Society of America, 2012, pp. AF4A–54.

[10] Y. Takita, K. Tajima, T. Hashiguchi, T. Katagiri, Wavelength defragmentation with minimum optical path disruptions for seamless service migration, in: Optical Fiber Communication Conference, Optical Society of America, 2016, pp. M2J–3.

[11] R.V. Fávero, J.S. Marçal, P.C. Silva, L.H. Bonani, M.L. Abbade, A new elastic optical network defragmentation strategy based on the reallocation of lightpaths sharing the most fragmented link, in: 2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference, IMOC, IEEE, 2015, pp. 1–5.

[12] M. Zhang, C. You, H. Jiang, Z. Zhu, Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic, J. Lightwave Technol. 32 (5) (2014) 1014–1023.

[13] T. Panayiotou, K. Manousakis, S.P. Chatzis, G. Ellinas, A data-driven bandwidth allocation framework with QoS considerations for EONs, J. Lightwave Technol. 37 (9) (2019) 1853–1864.

[14] P. Morales, P. Franco, A. Lozada, N. Jara, F. Calderón, J. Pinto-Ríos, A. Leiva, Multi-band environments for optical reinforcement learning gym for resource allocation in elastic optical networks, in: 2021 International Conference on Optical Network Design and Modeling, ONDM, 2021, pp. 1–6.

[15] S. Troia, A. Rodriguez, I. Martín, J.A. Hernández, O.G. De Dios, R. Alvizu, F. Musumeci, G. Maier, Machine-learning-assisted routing in SDN-based optical networks, in: 2018 European Conference on Optical Communication, ECOC, 2018, pp. 1–3.

[16] S. Trindade, N.L. da Fonseca, Machine learning for spectrum defragmentation in space-division multiplexing elastic optical networks, IEEE Netw. 35 (1) (2020) 326–332.

[17] Y. Xiong, Y. Yang, Y. Ye, G.N. Rouskas, A machine learning approach to mitigating fragmentation and crosstalk in space division multiplexing elastic optical networks, Opt. Fiber Technol., Mater. Devices Syst. 50 (2019) 99–107.

[18] S. Troia, R. Alvizu, G. Maier, Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks, IEEE Access PP (2019) 1.

[19] X. Wang, Q. Zhang, I. Kim, P. Palacharla, M. Sekiya, Utilization entropy for assessing resource fragmentation in optical networks, in: Optical Fiber Communication Conference, Optical Society of America, 2012, pp. OTh1A–2.

[20] P. Wright, M.C. Parker, A. Lord, Minimum-and maximum-entropy routing and spectrum assignment for flexgrid elastic optical networking, J. Opt. Commun. Netw. 7 (1) (2015) A66–A72.

[21] M. Zhang, W. Shi, L. Gong, W. Lu, Z. Zhu, Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions, in: 2013 IEEE International Conference on Communications, ICC, IEEE, 2013, pp. 3894–3898.

[22] Y. Wang, J. Zhang, Y. Zhao, J. Liu, W. Gu, et al., Spectrum consecutiveness based routing and spectrum allocation in flexible bandwidth networks, Chin. Optics Lett. 10 (s1) (2012) S10606.

[23] P. Liashchynskyi, P. Liashchynskyi, Grid search, random search, genetic algorithm: A big comparison for NAS, 2019, arXiv.

[24] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, S.J.B. Yoo, Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks [invited], J. Opt. Commun. Netw. 5 (10) (2013) A100–A106.

[25] Q.H. Nguyen, H.-B. Ly, L.S. Ho, N. Al-Ansari, H.V. Le, V.Q. Tran, I. Prakash, B.T. Pham, Influence of data splitting on performance of machine learning models in prediction of shear strength of soil, Math. Probl. Eng. 2021 (2021) 4832864.

[26] R. Livni, S. Shalev-Shwartz, O. Shamir, On the computational efficiency of training neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates, Inc., 2014.

[27] H. Meunier, E.-G. Talbi, P. Reininger, A multiobjective genetic algorithm for radio network optimization, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), Vol. 1, 2000, pp. 317–324.

[28] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, L. Salomon, Performance indicators in multiobjective optimization, European J. Oper. Res. 292 (2) (2021) 397–422.