# Deep Learning for Speeding up the Min Slot-Continuity Capacity Loss Spectrum Assignment

1st Matheus L. Santos
*Federal University of Pernambuco - Brazil*
matheus.lobo@ufpe.br

2nd José Hélio da C. Júnior
*Federal University of Pernambuco - Brazil*
joseheliodacruzjunior@gmail.com

3rd Karcius D. Rosário Assis
*Federal University of Bahia - Brazil*
karcius.assis@ufba.br

4th Raul C. Almeida Jr
*Federal University of Pernambuco - Brazil*
raul.almeidajunior@ufpe.br

5th Raouf Boutaba
*University of Waterloo - Canada*
rboutaba@uwaterloo.ca

2023 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC) | 979-8-3503-2067-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/IMOC57131.2023.10379781

***Abstract*—This paper presents two classifier models based on deep neural networks to speed up the Min Slot-Continuity Capacity Loss (MSCL) spectrum assignment. The first decides between the use of First-Fit or MSCL heuristic, with the aim of avoiding unnecessary MSCL calls whenever the application of First-Fit would provide the same minimum loss of capacity as MSCL. The second adds the capability of pointing out the correct portion of the spectrum MSCL should look for whenever First-Fit is not selected. Simulation results demonstrate reductions of 28% and 74% on the simulation time between the MSCL and the two proposed models without mitigation on the MSCL performance.**

***Index Terms*—Elastic Networks, MSCL, Deep Neural Network.**

## I. Introduction

RSA problems applied to optical networks are computationally expensive, being classified as an NP-hard problem. This classification implies that the optimal solution to the problem cannot be obtained in an acceptable time when applied to large scenarios [1]. Therefore, heuristics and meta-heuristics are developed and studied to find a solution to the problem. This process ends up generating small fragments of spectrum that, when they do not accommodate the size of the requested demands, cause spectrum underutilization.

To mitigate fragmentation problems, [2] proposed a spectrum assignment heuristic called Min Slot-Continuity Capacity Loss (MSCL). The MSCL receives a slot-availability vector with the slots' status (available or not), and calculates the reduction in assignment capacity that would occur if a specific group of contiguous slots were chosen to attend the request. This is performed for each interfering route, and the set of slots that provides the lowest capacity reduction is chosen.

In the works [3] and [4], comparisons were made between the MSCL and First-Fit (FF) heuristics for scenarios of elastic optical networks. The results of these works demonstrate that the MSCL heuristic, when compared to the FF, presents significant reductions in the path-request blocking probability. However, MSCL demands a lot of processing time to search the solution. Therefore, the time required to execute the MSCL is orders of magnitude greater than that of FF.

In a spectrum-assignment comprehensive study, it was observed that when MSCL is applied to the single slot-availability scenario, MSCL chooses the same spectrum assignment as FF for about 60% of requests. Thus, knowing these moments, it is possible to avoid the calculation of MSCL when using the FF spectrum assignment policy and provide exactly the same spectrum choice. This article proposes the creation of supervised deep learning models to create an engine that is capable of taking efficient decisions as the MSCL does, but in reduced processing times. This article focus on the calculation of the loss of capacity on a single slot-availability vector. This is appropriate for a point-to-point scenario or in networks under Full-Spectrum Spatial Switching, as proposed in [5]. Future works shall extend this work for other network topology scenarios.

## II. Default MSCL Heuristic

In [2], the MSCL spectrum assignment heuristic was presented. The objective of the MSCL is to analyze and returns the set of slots that provide the lowest aggregated capacity loss, for each request, the state of the possible optical paths in the network in order to obtain the best set of slots to allocate the request. Which is the one with the least capacity loss. This choice effectively reduces the blocking of subsequent requests. The way how MSCL calculates the capacity loss is based on the number of possible forms how to allocate a request in the spectrum gaps (set of contiguous available slots). Let $n$ be the number of possible upcoming requested slots (demands) and $v$ the number of consecutive available slots in a spectrum gap. If $n \geq v$, the number of possible shapes to allocate the request is $n - v + 1$; and zero otherwise. The process to obtaining the lowest capacity loss is performed considering an allocation in all valid slots of the spectrum. The set of slots that presents the lowest capacity loss is returned by the algorithm as the best allocation for a given demand.

## III. Proposed Models

This article presents classification models based on deep learning capable of reducing the processing burden of the MSCL spectrum allocation algorithm. Theses models avoid unnecessary calls to capacity loss calculation at the same time that MSCL efficient spectrum range decisions are conducted. Two models have been developed, which are compared to the standard FF and MSCL in blocking probability and execution time. The models are named 'Binary Model' and 'Multi Class (1,4) Model', the latter with two versions. Throughout this session, we detail the models and their hyperparameter settings used in the training process.

### A. Binary Model

The first model is a binary classifier built from a 4-layer feedforward neural network. The deep neural network receives as input a binary S-size vector with the occupation of each slot in addition to the requested number of slots, $n$. The output of the neural network switches between FF and MSCL heuristics. The binary value '0' indicates FF, while '1' indicates MSCL. In order to obtain the data for training the neural network, simulations were performed using the MSCL as the heuristic decision. The state of the network before each allocation, $S$, the requested number of slots, $n$, and the correct label of the classifier are saved. To obtain the label, it is necessary to observe the initial slot index informed by the MSCL. If it coincides with FF, label 0 is assigned. Otherwise, label 1 is assigned.

A total of 5 million of samples distributed using network load from 50 to 65 Erlangs were collected. The data are correct balancing between the two classes, i.e., FF and MSCL have the same distribution. The dataset were divided into 2 sets for training and testing. The validation set has $200,000$ samples. In this stage of data pre-processing, the manipulation of category variables for 'one-hot encoding' was carried out. Thus, the demands can assume three values being encoded by 100, 010 and 001. These are added to the 320 slots that represent the availability of the spectrum, resulting in a neural network with 323 input bits.

The deep neural network was developed in Python using the PyTorch framework. A 4-layer architecture with 323, 512, 128 and 64 neurons per layer, respectively, was defined after convergence tests. The activation function considered in each hidden layer is a rectified linear unit (ReLU) function, meanwhile the output layer uses a Sigmoid function. The Binary Cross Entropy Loss (BCELoss) function was used, since it is an optimized function for binary classification. The Adam optimizer was used with an initial learning rate ($LR$) of $10^{-3}$. In addition, the $L_2$ regularization (Weight Decay) of $10^{-3}$ was used to prevent overfitting.

### B. Multi Class (1,4) Model v1

This model extends the binary model to new classes. According to [4], the number of slots analyzed by MSCL increases the execution time of the heuristic. Subdividing the spectrum into only on the defined region and performing the capacity loss calculation would mitigate the execution time of MSCL. We propose that the process of identifying which region will be used be performed by a deep neural network. Then, the proposed model classifies output into five distinct classes. Similar to the binary model, data is collected by assigning label '0' whenever MSCL indicates the same allocation as FF. Whenever FF is not chosen, the other classes indicate the region of the spectrum in which MSCL must carry out its allocation. This procedure is performed by subdividing the frequency spectrum into quadrants and observing the first slot allocated to the demand. With $x$ being the first allocation slot and using 320 frequency slots, (1) indicates the following spectrum slice for the label.

$$\text{slice} = \begin{cases} 1 & \text{if } 0 \leq x < 80 \\ 2 & \text{if } 80 \leq x < 160 \\ 3 & \text{if } 160 \leq x < 240 \\ 4 & \text{if } 240 \leq x < 320 \end{cases} \tag{1}$$

The data pre-processing was carried out in the same way as the binary model. However, a new dataset with 2,775,000 samples was created due to the new classes. The dataset has a fair balancing between the five classes. Because it is a multiclass problem, some changes were made. The amount of neurons in the hidden layers was increased, resulting in a neural network with 323, 512, 256 and 128, in addition to 5 outputs indicating the distributions of each class being selected. The ReLU activation function was maintained for the hidden layers. However, the output is transformed with the Softmax function. Dropout layers were placed between each hidden layer, with a dropout probability of $20\%$. The loss function was changed to Cross Entropy Loss because its optimization work with multiple classes. The Adam optimizer was kept with learning rate of $10^{-4}$ and $L_2$ of $10^{-3}$.

### C. Multi Class (1,4) Model v2

The second version of the multiclass model follows the premise of its predecessor, where the main difference is in the size of the deep neural network. Seeking to visualize the impact of a robust neural network, adjustments were made to the hyperparameters. The number of layers was increased by one unit, totalizing 5 layers, with 323, 1024, 512, 256 and 128 neurons, respectively. The $L_2$ coefficient was changed to $5 \times 10^{-4}$ and the other settings were maintained. Whenever the region chosen by the multiclass models cannot perform the allocation, the default MSCL for the entire spectrum is called. The same approach applies in version one.

## IV. Results

To evaluate the performance of the proposed models, computational experiments were carried out aided by an elastic optical network simulator. The simulator used was written in Python and checked by making comparisons with the simulator SIMTON [6]. The spectrum contains 320 frequency slots, and this value is obtained by following commonly used values in the literature. For each simulation, $5 \times 10^5$ requests were performed, executed 3 times and averaged. The incoming traffic assumes a Poissonian process and the request duration follows an Exponential distribution. The load is given by dividing the average connection holding time by the average inter-arrival time. To obtain the possible demands for slots, 8-QAM modulation format and rates of 100, 200 and 400 Gbits/s were considered. The calculation of the size of the demands follows the equation presented in [7], which results in demands of 2, 3 and 6 slots for the respective rates used.

The accuracy of the binary model was 87.15%. The model's accuracy directly impacts the blocking probability and simulation time. Choosing MSCL when the FF could be employed only degrades the simulation time. On the other hand, a wrong choice of FF instead of MSCL may result in not proper assignment. Models with multiple classes obtained an accuracy of 72.44% and 85.64%, for the versions 1 and 2, respectively. This difference is reflected in the blocking probability and simulation time results.

Fig. 1(a) presents the blocking probabilities (BP) as a function of network load. It is important to note that the performance in BP of the three proposed models are close to the Defaul MSCL, which has been used as a benchmark. In Tab. I, it can be seen that the BP of Multi Class $(1, 4)$ Model v2 is lower than the other models'. The performance in BP of the Multi Class $(1, 4)$ Model v2 is superior to the presented by the other models. Differences to MSCL on the BP as low as 2.27% and $-3.26\%$ have been achieved with the use of Multi Class $(1, 4)$ Model v2. The accuracy is crucial in this comparison, as a wrong choice of region may prevent future allocations.

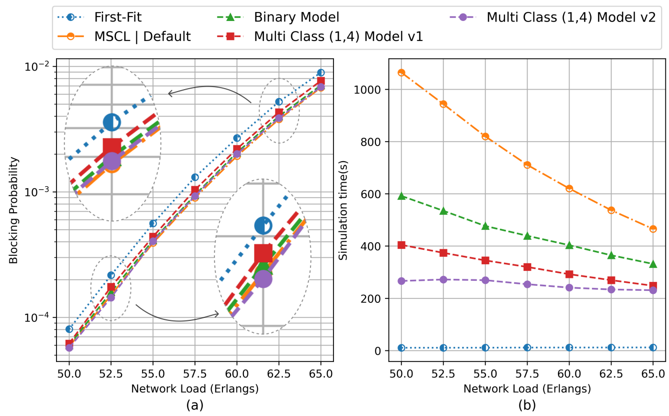


Fig. 1. (a) Blocking probability vs Network load. (b) Simulation time in seconds vs Network load.

Regarding the simulation time, presented in Table II, we can observe the resulting mitigation in processing time with the adoption of the three proposed models. For a load of 50 Erlangs, Multi Class (1,4) Model v2 showed a reduction of 74.95% compared to the time required by the default MSCL. Network load increase impacts the execution time reduction of the three proposed models. This occurs because, at lower loads, there is less spectrum occupation and the MSCL needs to analyze a larger number of slots. A very interesting observation is that even with a larger neural network, Multi Class (1,4) Model v2 presents the lowest execution time among all strategies. When compared to Multi Class (1,4) Model v1, this occurs due to its higher accuracy in matching the correct regions of the spectrum.

TABLE I
BLOCKING PROBABILITY REDUCTION COMPARED TO MSCL

| Model | Network Load | | |
|---|---|---|---|
| | **50.0** | **57.5** | **65.0** |
| First-Fit | -37.50% | -45.77% | -31.99% |
| MSCL — Default | - | - | - |
| Binary Model | -2.27% | -5.12% | -4.46% |
| Multi Class (1,4) Model v1 | -5.68% | -15.73% | -13.40% |
| Multi Class (1,4) Model v2 | 2.27% | -3.26% | -1.37% |

TABLE II
EXECUTION TIME REDUCTION COMPARED TO MSCL

| Model | Network Load | | |
|---|---|---|---|
| | **50.0** | **57.5** | **65.0** |
| First-Fit | 98.98% | 98.37% | 97.40% |
| MSCL — Default | - | - | - |
| Binary Model | 44.37% | 38.17% | 28.71% |
| Multi Class (1,4) Model v1 | 61.99% | 54.96% | 46.68% |
| Multi Class (1,4) Model v2 | 74.95% | 64.26% | 50.33% |

## V. Conclusion

This paper presents two deep learning models that keep the same spectrum allocation efficiency as the MSCL, but with significant reductions on the execution time. Simulation times as low as 75% of that provided by MSCL were achieved, proving the effectiveness of the proposed models.

## Acknowledgment

We would like to thank CNPq for the financial support provided as well as all universities who have assisted the publication.

## References

[1] K.-T. Ko, K.-S. Tang, C.-Y. Chan, K.-F. Man, and S. Kwong, "Using genetic algorithms to design mesh networks," *Computer*, vol. 30, no. 8, pp. 56–61, 1997.

[2] R. C. Almeida-Jr., A. F. d. Santos, K. Assis, H. Waldman, and J. Martins-Filho, "Slot assignment strategy to reduce loss of capacity of contiguous-slot path requests in flexible grid optical networks," *Electronics Letters*, vol. 49, no. 5, pp. 359–361, 2013.

[3] M. L. d. Santos, R. C. Almeida Jr, and D. R. Araújo, "Heurísticas para atribuição de espectro em redes ópticas elásticas baseadas em perda de capacidade sob múltiplas rotas," Sept. 2021.

[4] M. L. d. Santos, "Abordagens para atribuição de espectro em redes ópticas elásticas baseadas em perda de capacidade sob múltiplas rotas," Master's thesis, Universidade Federal de Pernambuco, 2021.

[5] A. C. Jatoba-Neto, D. A. Mello, C. E. Rothenberg, S. Ö. Arik, and J. M. Kahn, "Scaling sdm optical networks using full-spectrum spatial switching," *Journal of Optical Communications and Networking*, vol. 10, no. 12, pp. 991–1004, 2018.

[6] D. A. Chaves, H. A. Pereira, C. J. Bastos-Filho, and J. F. Martins-Filho, "Simton: A simulator for transparent optical networks," *Journal of Communication and Information Systems*, vol. 25, no. 1, 2010.

[7] M. M. Alves, R. C. Almeida, A. F. dos Santos, H. A. Pereira, and K. D. Assis, "Impairment-aware fixed-alternate bsr routing heuristics applied to elastic optical networks," *The Journal of Supercomputing*, vol. 77, pp. 1475–1501, 2021.