

UNIVERSIDAD NACIONAL DE ASUNCIÓN

Facultad Politécnica



“ESTRATEGIA DE DISPARO PARA EL PROCESO DE  
DESFAGMENTACIÓN EN REDES ÓPTICAS ELÁSTICAS,  
UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO”

TRABAJO FINAL DE GRADO PRESENTADO POR

JOSÉ LUIS ENCISO DACAK  
Y NICOLÁS ALEJANDRO SILVA CORONEL

COMO REQUISITO  
PARA OBTENER EL TÍTULO DE INGENIERO  
EN INFORMÁTICA

ORIENTADOR:  
PHD. MSc. ING. ENRIQUE DAVALOS

San Lorenzo - Paraguay.

Noviembre de 2021

# Índice general

Índice de Figuras	III
Índice de Tablas	IV
Lista de Símbolos	v
<b>1. Introducción</b>	<b>1</b>
1.1. Justificación . . . . .	1
1.2. Objetivos del trabajo . . . . .	4
1.2.1. Objetivo General . . . . .	4
1.2.2. Objetivos Específicos . . . . .	4
1.3. Organización del libro . . . . .	4
<b>2. Fragmentación en Redes Ópticas Elásticas</b>	<b>6</b>
2.1. Fragmentación del Ancho de Banda en EON . . . . .	7
2.1.1. Enfoques de gestión de fragmentación . . . . .	9
2.2. Descripción del problema tratado . . . . .	11
2.3. Análisis Bibliográfico . . . . .	12
<b>3. Aprendizaje Automático</b>	<b>17</b>
3.1. Tipos de sistemas de aprendizaje de máquinas . . . . .	18
3.2. Redes Neuronales Artificiales . . . . .	19
3.2.1. Perceptrón . . . . .	19

3.2.2. Funciones de activación . . . . .	21
3.2.3. Arquitectura y aprendizaje . . . . .	22
3.3. Aplicación de Machine Learning en Redes Ópticas Elásticas . . . . .	24
<b>4. Método Propuesto</b>	<b>27</b>
4.1. Características . . . . .	27
4.2. Obtención de datos para el entrenamiento . . . . .	30
4.3. Herramientas Utilizadas . . . . .	32
4.4. Modelado . . . . .	33
4.5. Entrenamiento . . . . .	33
4.6. Pruebas de predicción . . . . .	34
<b>5. Pruebas y resultados obtenidos</b>	<b>36</b>
5.1. Objetivos a optimizar . . . . .	38
5.2. Análisis de los resultados . . . . .	38
<b>6. Conclusiones y Trabajos Futuros</b>	<b>44</b>
6.1. Conclusiones Experimentales . . . . .	45
6.2. Aportes . . . . .	46
6.3. Trabajos Futuros . . . . .	47

# Índice de figuras

2.1. Redes WDM v s Redes EON . . . . .	7
2.2. Restricciones de contigüidad y continuidad en redes EON . . . . .	8
2.3. Esquema de Gestión de la Fragmentación . . . . .	9
2.4. Ejemplo de desfragmentaciones periódicas con volumen de carga de tráfico variado . . . . .	12
2.5. Algoritmo propuesto por Favero y colegas [12] . . . . .	15
3.1. Comportamiento del perceptrón . . . . .	20
3.2. Funciones de Activación . . . . .	21
3.3. Ejemplo de red neuronal . . . . .	22
4.1. Ejemplo de métricas de fragmentación . . . . .	31
4.2. Volumen de tráfico variado utilizado . . . . .	32
4.3. Evolución del error a través de épocas . . . . .	34
4.4. Utilización de la red y predicciones . . . . .	35
5.1. Diagrama simulador / modelo entrenado . . . . .	37
5.2. Gráfico de soluciones en el frente pareto para la topología NSFNET . .	40
5.2. Gráfico de soluciones en el frente pareto para la topología NSFNET . .	41
5.3. Gráfico de soluciones en el frente pareto para la topología USNET . .	41
5.3. Gráfico de soluciones en el frente pareto para la topología USNET . .	42

# Lista de Tablas

4.1. Tabla de resultados en pruebas de predicción . . . . .	34
5.1. Tabla de soluciones en el frente pareto . . . . .	39
5.2. Tabla de cobertura para topología USNET . . . . .	43
5.3. Tabla de cobertura para topología NSFNET . . . . .	43

# Lista de Símbolos

$HM$	<i>High-slot Mark.</i>
$HM_{max}$	HM máximo.
$Ent_{link}$	Entropía del enlace.
$Ent_{link-i}$	Entropía del enlace $i$ .
$Ent_{red}$	Entropía de la red.
$N$	Cantidad de $FS$ en un enlace.
$FS_i$	<i>Frecuency Slot</i> de índice $i$ .
$FS_{i+1}$	<i>Frecuency Slot</i> de índice $i + 1$ .
$ E $	Cantidad de enlaces de la red.
$SHF_{link}$	Entropía de Shannon del enlace.
$SHF_{link-i}$	Entropía de Shannon del enlace $i$ .
$SHF_{red}$	Entropía de Shannon de la red.
$S_{free}$	Cantidad de $FS$ libres en un enlace.
$BFR_{link}$	Relación de Fragmentación de ancho de banda de un enlace.
$BFR_{link-i}$	Relación de Fragmentación de ancho de banda del enlace $i$ .
$BFR_{red}$	Relación de Fragmentación de ancho de banda de la red.
$MaxBlock()$	Tamaño del mayor bloque de $FS$ bloqueados.
$MSI$	Índice de slot máximo utilizado.
$MSI_{link-i}$	Índice de slot máximo utilizado del enlace $i$ .
$MSI_{red}$	Índice de slot máximo utilizado de la red.

$CE_{link}$	Consecutividad del espectro.
$CE_{red}$	Consecutividad del espectro de la red.
$Joins$	Cantidad total de bloques de dos ranuras libres adyacentes distintos dentro de un enlace.
$Bloques$	Cantidad de bloques de ranuras libres en un enlace.
$K$	Cantidad de rutas de dos enlaces en la red.
$Uso$	Porcentaje de utilización de la red.
$sum(i)$	Cantidad de $FS$ utilizadas en el enlace $i$ .
$FSB$	Acumulación de $FS$ bloqueados.
$S_i^{block}$	Cantidad de $FS$ solicitadas por la demanda bloqueada $i$ .
$D$	Cantidad de demandas.
$T$	Ventada de tiempo seleccionada para el calculo de $IB$ .
$PB_t$	Índice de bloqueo para el tiempo $t$ .
$FSD_i$	Cantidad de $FS$ demandadas en el tiempo $t$ .
$train_{mean}$	Media de valores.
$train_{stf}$	Desviación estándar.
$MAE$	Error Absoluto Medio.
$MSE$	Error Cuadrático Medio.
$PB$	Probabilidad de Bloqueo.
$PB_{th}$	Umbral para disparar el proceso de desfragmentación.
$BL$	Cantidad de Bloqueos.
$RC$	Cantidad de Reconfiguraciones.
$SFP$	Número de soluciones en el Frente Pareto.
$CP$	Cobertura Pareto.
$MP$	Método Propuesto.

# Capítulo 1

## Introducción

### 1.1. Justificación

El amplio crecimiento de las demandas de tasas de bits en las redes de computadoras debido al incremento de la popularidad de la Internet y servicios basados en las nubes, tales como *Content Delivery Network* (CDN) y *Video on Demand* (VoD) obligan a estudiar nuevas y mejores tecnologías relacionadas a la transmisión de datos [1].

Las Redes de Multiplexación por División de Longitud de Onda o *Wavelength Division Multiplexing* (WDM), utilizan una grilla fija, de 50 o 100 GHz, dan una gran ventaja logrando velocidades muy superiores frente a las viejas tecnologías, pero a pesar de esta ventaja señalada, la gruesa granularidad lleva a un uso ineficiente del espectro, ya que cada demanda es asignada a un canal fijo y estas pueden requerir un ancho de banda menor al tamaño del canal.

Esta desventaja da lugar a las Redes Elásticas Ópticas o *Elastic Optical Networks* (EON) [2], las cuales surgen como una solución al problema anteriormente citado, ya que estas proporcionan una mayor flexibilidad en la división del espectro y de esa forma lograr que los requerimientos sean asignados de manera más eficiente.

A las redes EON también se la conocen como redes de grilla flexible, debido a que las ranuras de frecuencia o FS (*Frequency Slot*) que reemplazan a los “Canales



WDM”, cuentan con una división más flexible. Cada FS tiene un ancho de banda de 12.5 GHz, de esta manera se logra una cantidad más apropiada de FS para satisfacer un requerimiento.

Los métodos de ruteo y asignación del espectro tienen gran impacto sobre el uso eficiente de los recursos de la red. Los algoritmos RSA (*Routing and Spectrum Assignment*) se encargan de resolver dicho problema encontrando el camino más apropiado desde el origen hasta el destino y las ranuras que utilizará el requerimiento dentro del espectro de los enlaces.

Se han propuestos varios algoritmos RSA con el fin de conseguir la mejor utilización de recursos, estos algoritmos están sujetos a dos principios fundamentales: la restricción de consecutividad del ancho de banda y la restricción de la continuidad del ancho de banda [3].

La restricción de continuidad establece que se deben utilizar los mismos FS en todo el camino y la restricción de contigüidad dispone que los FS seleccionados para satisfacer la demanda deben ser contiguos.

Debido a las restricciones explicadas y a que las asignaciones de recursos son realizadas de manera dinámica, surge el fenómeno denominado "Fragmentación del Ancho de Banda", este problema es una de las principales dificultades de las redes EON ya que tiene un impacto directo en el uso eficiente del espectro.

El fenómeno de la fragmentación del ancho de banda sucede cuando en los enlaces se encuentran FS disponibles separados por FS que están siendo utilizados por otras conexiones, por lo que estas podrían quedar inutilizables para nuevas conexiones por no poder satisfacer a la demanda debido a las restricciones citadas anteriormente, en consecuencia, la probabilidad de bloqueo [4] aumenta considerablemente.

Un bloqueo sucede cuando el algoritmo RSA no puede encontrar FS disponibles para una demanda, esto puede deberse a una alta saturación del espectro, pero también debido al problema mencionado anteriormente, donde existe la cantidad de FS libres que se solicitan, pero sin respetar las restricciones de continuidad y

contigüidad, es decir el espectro se encuentra fragmentado.

El problema de la fragmentación de redes EON es ampliamente estudiado en la literatura actual, para buscar manejarlo se han propuesto soluciones con distintos enfoques.

Uno de los enfoques es el llamado *Enfoque proactivo* el cual consiste en ejecutar un proceso de desfragmentación periódicamente o mediante un disparador. Tiene como principal objetivo prevenir futuros bloqueos en la red, este enfoque será el utilizado en este trabajo.

El proceso de desfragmentación consiste en la reconfiguración o re-ruteo de un sub-conjunto de conexiones ya establecidas en la red, teniendo como principal objetivo reducir la fragmentación del espectro mediante la eliminación de bloques de FS libres no contiguos.

En el trabajo presentado por Zhang [5], se realizó un análisis del problema de desfragmentación en redes EON, en el cual lo dividen en cuatro subproblemas, los cuales son, (I) ¿Cómo reconfigurar?, (II) ¿Cómo migrar el tráfico?, (III) ¿Cuándo reconfigurar? y (IV) ¿Qué reconfigurar?.

En este trabajo nos centraremos en el tercer subproblema, ¿Cuándo reconfigurar?, ya que considerando el enfoque proactivo para resolver el problema de la fragmentación, encontramos que los procesos de desfragmentación podrían ejecutarse en periodos de tiempo donde no son del todo necesarios, es decir cuando la red se encuentra con una baja fragmentación, provocando desfragmentaciones ineficientes, una cantidad mayor de interrupciones de conexiones y una elevación innecesaria del costo de procesamiento.

En los siguientes capítulos presentamos un novedoso modelo de predicción de probabilidades de bloqueo implementado con técnicas de aprendizaje automático o *Machine Learning*, el cual se utiliza como disparador del proceso de desfragmentación, proponiendo de esta manera una solución al sub problema planteado anteriormente

## 1.2. Objetivos del trabajo

### 1.2.1. Objetivo General

Diseñar un modelo de disparo para el proceso de desfragmentación en redes ópticas elásticas basado en probabilidades de bloqueo, utilizando técnicas de aprendizaje automático o *Machine Learning*, con el propósito de conseguir una mayor eficiencia en el uso de los recursos de la red, realizando una menor cantidad de reconfiguraciones de conexiones existentes y minimizando la cantidad total de bloqueos obtenidos.

### 1.2.2. Objetivos Específicos

- Realizar un análisis bibliográfico del estado del arte.
- Establecer características o parámetros claros que representen el grado de fragmentación de la red.
- Desarrollar un algoritmo con herramientas de redes neuronales que permita entrenar nuestro modelo con datos previamente recolectados.
- Implementar una interfaz entre el simulador de redes EON y el modelo de redes neuronales entrenado a fin de establecer un canal de comunicación entre ambas.
- Realizar pruebas de nuestro método y compararlas con otras técnicas ya existentes.

## 1.3. Organización del libro

El presente trabajo se encuentra organizado de la siguiente manera:

En el capítulo dos se trata sobre características y conceptos relacionados con las redes EON, su principal dificultad (la fragmentación del ancho de banda), los dife-

rentes enfoques para manejar la misma y una presentación de trabajos relacionados presentes en la literatura científica.

En el capítulo tres se hace una introducción al *Machine learning*, enfocado al aprendizaje supervisado y redes neuronales.

En el capítulo cuatro se presenta el método propuesto para la selección del momento de desfragmentación, describiendo todo el proceso que conlleva.

El capítulo cinco se muestra las pruebas experimentales junto a un análisis de los resultados obtenidos.

Por último, el capítulo seis presenta las conclusiones del trabajo y sugerencias para trabajos futuros.

## Capítulo 2

# Fragmentación en Redes Ópticas Elásticas

Las redes ópticas tradicionales basadas en WDM, también conocidas como redes ópticas de grilla fija o *fixed-grid networks*, dividen el espectro de cada enlace en canales fijos de 50 GHz o 100 GHz de ancho de banda, según se especifica en el estándar G.694.1 del año 2002 de la Unión Internacional de Telecomunicaciones (*ITU-T International Telecommunication Union*).

Debido a la rigidez de estas redes WDM, es posible que ocurra una utilización ineficiente de la capacidad, esto es debido a que el espacio entre dos canales adyacentes es relativamente grande y si la señal transmitida usa un ancho de banda bajo, gran parte del espectro será desperdiciado.

Una nueva tecnología denominada Redes Ópticas Elásticas o *Elastic Optical Networks* (EON) propuestas por Masahiko Jinno en [2], también denominadas como redes ópticas de grilla flexible o *Flex-grid Networks*, dividen el espectro óptico en Ranuras de Frecuencia o *Frequency Slots* (FS) de 12.5 GHz de acuerdo con el estándar definido en ITU-T (G.694.1) en el año 2012.

En la Figura 2.1, se muestra una comparación en la asignación de demandas en ambas tecnologías, observándose un mejor aprovechamiento del espectro óptico para el caso de las redes EON, debido a un menor desperdicio de ancho en banda.

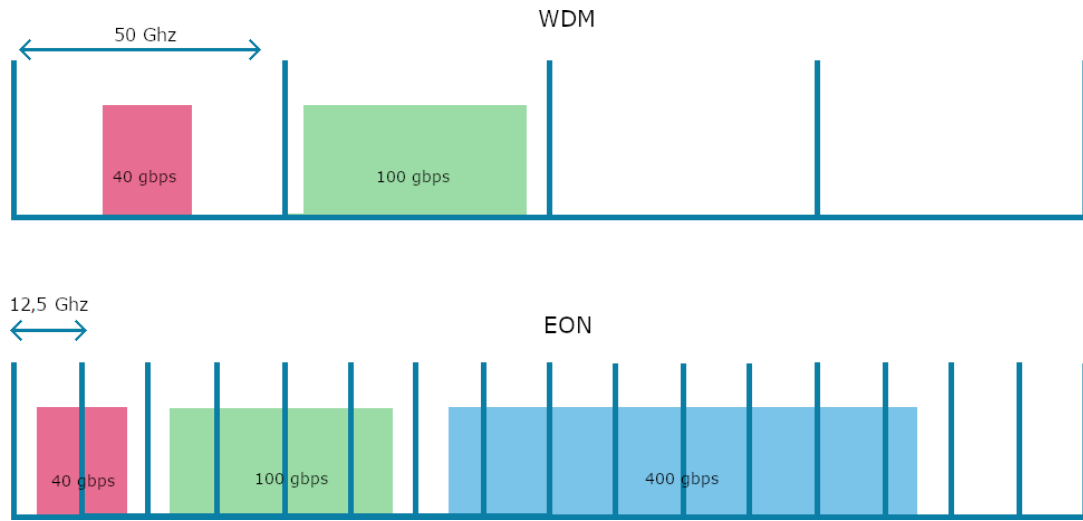


Figura 2.1: Redes WDM v s Redes EON

## 2.1. Fragmentación del Ancho de Banda en EON

Las redes elásticas ópticas permiten que el ancho de banda requerido por la demanda sea ubicado en el lugar más conveniente, siguiendo dos principales restricciones:

- **Restricción de continuidad del espectro:** los recursos de ancho de banda asignados a cada demanda deben ser los mismos a través de todos los enlaces físicos de la ruta.
- **Restricción de consecutividad del espectro:** el ancho de banda asignado a una demanda debe ser contiguo en el espectro, formando un solo bloque de FS.

Estas restricciones dan lugar a que luego de varias asignaciones de demandas y liberaciones de recursos se produzca la aparición de bloques aislados de FS no utilizados en los enlaces ópticos y no coincidentes unos respecto a otros en enlaces consecutivos, en consecuencia, esto permite una alta probabilidad de bloqueo, por lo que la red podría no satisfacer una demanda a pesar de tener ancho de banda disponible en los enlaces, este problema se conoce como “Fragmentación de la red”.

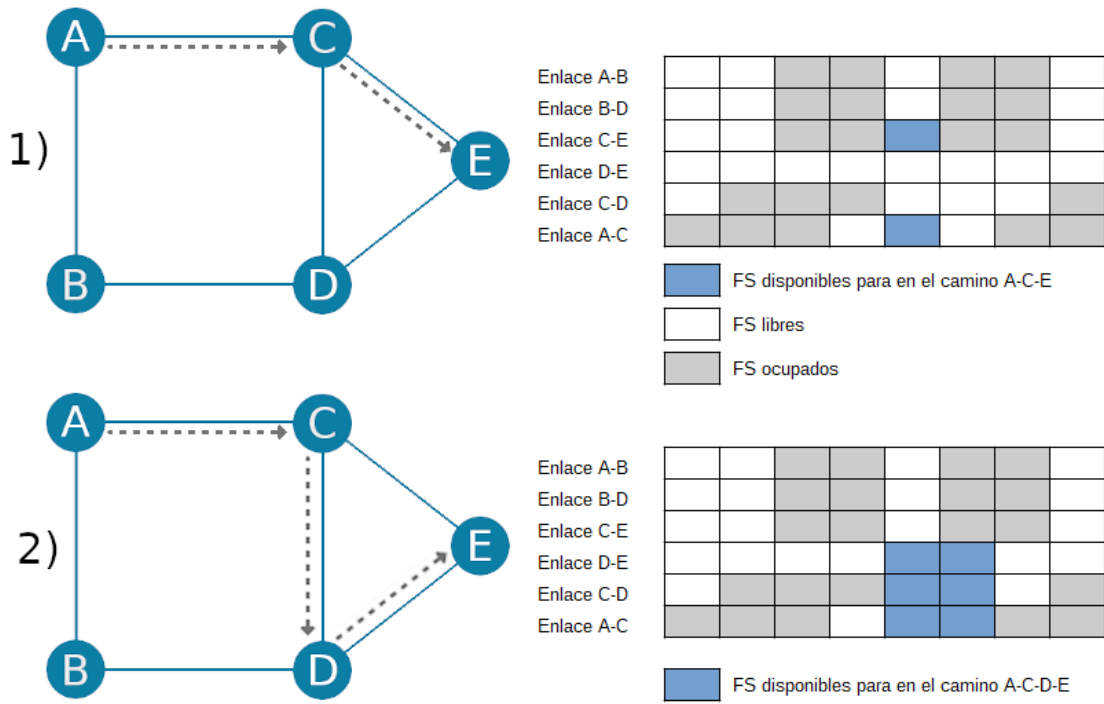


Figura 2.2: Restricciones de contigüidad y continuidad en redes EON

Un ejemplo de un escenario donde se pueden apreciar las restricciones de continuidad y contigüidad se muestra en la figura 2.2 donde se considera una demanda de origen en el nodo “A” y destino en el nodo “B” con dos FS a ser asignados en el espectro.

El camino más corto entre ambos nodos sería el A-C-E, pero este camino es rechazado por el algoritmo RSA, ya que los enlaces A-C y C-E no cuentan con suficientes FS contiguos y alineados para satisfacer la demandas, tal y como se ve en la parte 1 de la figura 2.2.

El siguiente camino que encuentra el algoritmo RSA es el A-C-D-E, a diferencia del anterior este camino sí cumple con ambas restricciones por lo que la demanda se asignaría correctamente en los FS respectivos, la parte 2 de la figura 2.2 expone dicha situación.

De esta manera es posible observar que quedan algunos bloques aislados, como los bloques de los enlaces A-C y sus enlaces adyacentes, en los cuales sus bloques

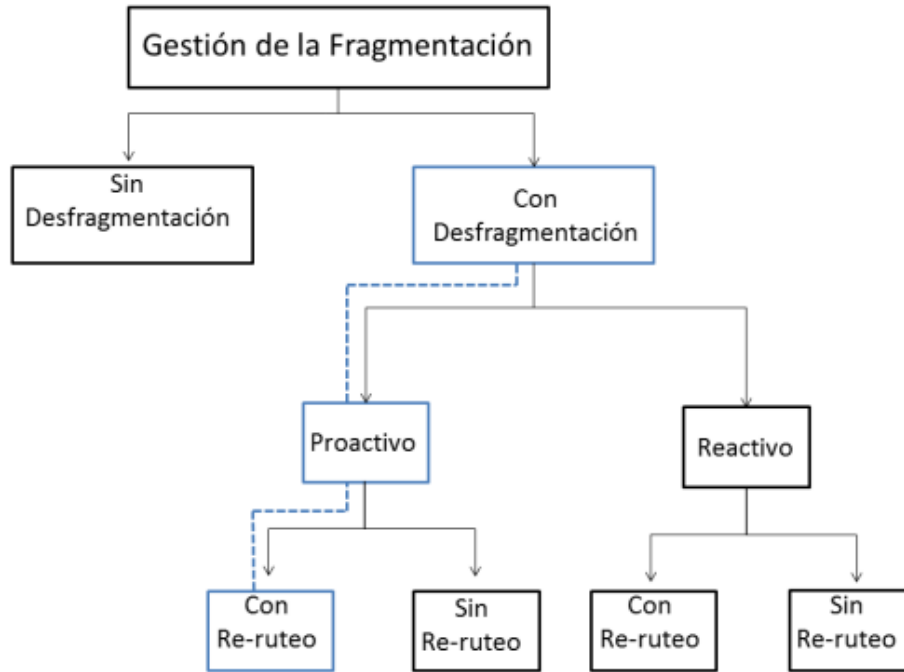


Figura 2.3: Esquema de Gestión de la Fragmentación

difícilmente sean utilizables ya que en su mayoría los bloques de un enlace no coinciden con el de los enlaces adyacentes. A esto es lo que llamamos fragmentación del espectro.

### 2.1.1. Enfoques de gestión de fragmentación

El problema explicado anteriormente trae consigo efectos adversos para la red, provoca que la probabilidad de bloqueo de la red aumente, afectando directamente a su correcto y fluido funcionamiento. Debido a esto es importante encontrar mecanismos que ayuden a evitar, alivianar o disminuir la fragmentación del ancho de banda.

Según la literatura científica existen diversos enfoques a ser considerados para la gestión de la desfragmentación, en la figura 2.3 se muestran los principales enfoques de gestión de la fragmentación [6].

La Desfragmentación es un proceso en el cual se realiza la reconfiguración o re-



ruteo de un sub-conjunto de conexiones existentes en la red, cuyo objetivo principal es reacomodar las asignaciones de espectro de las demandas de tráfico existentes para consolidar los espacios disponibles en grandes bloques contiguos y continuos que pueden ser utilizados para establecer futuras demandas [7].

Es posible gestionar el problema de la fragmentación sin utilizar técnicas de desfragmentación del espectro (Sin Desfragmentación), esto se logra realizando una gestión del espectro con el objetivo de evitar su fragmentación.

En el tratamiento de la fragmentación siguiendo un esquema Sin Desfragmentación podemos mencionar a los algoritmos denominados Sensibles a la Fragmentación o *Fragmentation Aware RSA* (FA-RSA). Los mismos consideran la fragmentación del ancho de banda para establecer las demandas siguiendo distintos indicadores de fragmentación y de esta manera se trata de minimizar la fragmentación del espectro.

Otra manera sería emplear técnicas de desfragmentación, utilizando dos enfoques principales:

- Desfragmentación Reactiva: El proceso se realiza ante el bloqueo de una demanda, con el objetivo de conseguir su establecimiento.
- Desfragmentación Proactiva: Se realiza periódicamente o considerando ciertos valores que disparen el proceso, de esta forma se logra disminuir la fragmentación de la red y se minimiza la aparición de futuros bloqueos de demandas.

Los enfoques que utilizan técnicas de desfragmentación también pueden clasificarse en: (i) enfoques sin re-ruteo, que realizan sólo una reasignación del espectro en los *lightpaths* o caminos de luz existentes, y (ii) con re-ruteos, que son técnicas que cambian las rutas y el espectro de los *lightpaths* existentes.

En el presente trabajo para la gestión de la fragmentación se utilizó el enfoque con desfragmentación, proactiva y con re-ruteos de *lightpaths* existentes, en la figura 2.3 se puede ver resaltado dicha estrategia.

## 2.2. Descripción del problema tratado

La Fragmentación del Ancho de Banda en Redes Ópticas Elásticas (EON) es un problema que afecta a la eficiencia en la utilización de recursos, el rendimiento de la red termina siendo profundamente afectada ya que este fenómeno podría provocar bloqueos de demandas por falta de ranuras disponibles contiguas y alineadas entre enlaces inmediatos y no necesariamente porque el espectro está totalmente ocupado. En secciones anteriores se explicaron estrategias para manejar la fragmentación de la red, en este trabajo se analiza la estrategia con desfragmentación, considerando un enfoque proactivo.

Un método ampliamente utilizado consiste en ejecutar el proceso de desfragmentación manera periódica con el fin de prevenir bloqueos futuros, abordando así una de las cuatro preguntas planteadas por Zhang [5], ¿Cuándo reconfigurar?.

En la figura 2.4 podemos ver una posible solución al problema de la selección del momento a realizar la desfragmentación, el cual consiste en realizar desfragmentaciones periódicas en tiempos fijos, en este caso cada 100 unidades de tiempo, el eje vertical indica el volumen de tráfico medido en el número de conexiones activas y el eje horizontal indica las unidades de tiempo, cada punto azul representa el momento en que el proceso de desfragmentación se ejecuta. Siguiendo este patrón se aprecia que se dan casos donde se realizan procesos de desfragmentación cuando la red podría no estar necesitando, si se considera que la utilización de la red es un indicador importante del grado de fragmentación.

Además de la utilización de la red, se tienen otras métricas de fragmentación importantes, cuyos valores deben tenerse en cuenta para el disparo de los procesos de desfragmentación.

De esta forma vemos la necesidad de un disparador inteligente para ejecutar el proceso de desfragmentación que considere todos estos parámetros o “características” para seleccionar convenientemente el momento del disparo, ya que realizar muchas

desfragmentaciones de manera frecuente afectan directamente al rendimiento de la red pudiendo causar disrupciones en las conexiones activas y en caso de realizar pocas y muy dispersas desfragmentaciones los efectos de las mismas serían casi imperceptibles.

En síntesis, la selección del momento para realizar el proceso de desfragmentación es crítica debido a que tiene un efecto importante en la cantidad de procesos de desfragmentación y esto tiene efecto en las dos métricas globales más importantes en el ruteo de redes elásticas las cuales son: Cantidad de bloqueos y Cantidad de reconfiguraciones.

En los siguientes capítulos presentamos y abordamos en profundidad un modelo de disparo inteligente, que tiene en cuenta numerosos factores tales como métricas de fragmentación de la red, utilización de la red y bloqueos de demandas.

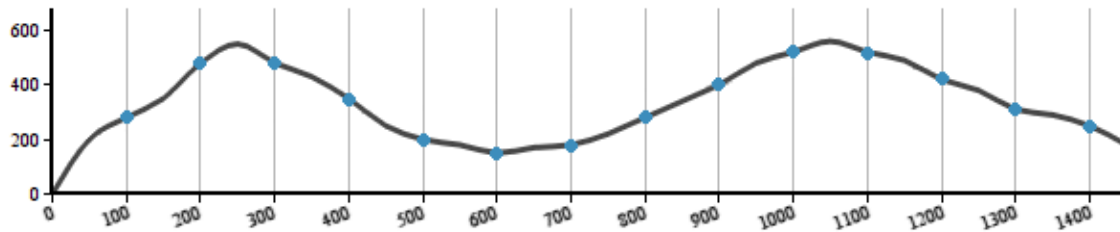


Figura 2.4: Ejemplo de desfragmentaciones periódicas con volumen de carga de tráfico variado

## 2.3. Análisis Bibliográfico

Seguidamente presentamos un análisis bibliográfico de trabajos presentes en el estado del arte que abordaron el mismo problema.

Jaume Comellas, Laura Vicario, y Gabriel Junyent [8] nos presentan un análisis de la desfragmentación periódica en redes EON con un tráfico dinámico, evaluando los diferentes efectos de los parámetros de desfragmentación en el rendimiento de la red.

El enfoque presentado en el trabajo consiste en realizar las desfragmentaciones en periodos fijos de  $N$  unidades de tiempo, con el principal objetivo de encontrar valores adecuados de  $N$ , ya que periodos de desfragmentación muy pequeños implican tener el espectro tan compactado como sea posible, pero a expensas de ejecutar el algoritmo con demasiada frecuencia, lo cual añade complejidad al proceso. Por otro lado, para valores muy altos de  $N$ , los efectos de la desfragmentación son insignificantes.

Este tipo de desfragmentaciones en periodos fijos es utilizado ampliamente en distintas investigaciones tales como [9], [10], entre otros. En nuestra investigación utilizamos esta técnica a fin de comparar los resultados con el disparador que proponemos.

Otra manera de afrontar el enfoque proactivo del proceso de desfragmentación es realizarlo mediante algún tipo de disparador de tal manera que la misma sea ejecutada solo en periodos de tiempo donde es realmente necesaria, a continuación, veremos algunos artículos los cuales usaron esta estrategia.

La investigación realizada por Yutaka Takita y colegas [11] propone un mecanismo de disparo para el proceso de desfragmentación basado en el valor del *High-slot Mark* (HM) el cual indica el número máximo de una ranura ocupada en la red. Utilizan esta métrica ya que lo consideran como una medida válida para evaluar la eficiencia en la utilización de los recursos. El proceso de desfragmentación se dispara de manera aleatoria cuando el valor del HM es mayor a un valor de  $HM_{max}$  definido previamente, en el artículo mencionado utilizan 30 como valor para  $HM_{max}$ .

Para el proceso de disparo en nuestro método propuesto se utilizan un conjunto de características o parámetros que indican el estado actual de la red, parte de ellas al igual que el *High-slot Mark* son también métricas que indican la fragmentación del espectro. En el capítulo 4 se explican en profundidad estas características donde una de ellas es el llamado *Maximum Slot Index* (MSI) el cual tiene una definición equivalente al del *High-slot Mark*.

Otra propuesta para el disparo es presentada por Ricardo V. Fávero y colegas

[12]. En su método combinan el enfoque reactivo y el enfoque proactivo para determinar el periodo en el que será ejecutado el proceso de desfragmentación, en la figura 2.5 se puede observar el diagrama que ilustra el algoritmo propuesto.

Inicialmente la variable  $d$  que utilizan para representar el estado de fragmentación se coloca en 0, el proceso de desfragmentación (DS) se ejecuta al cumplirse alguna de las siguientes condiciones.

Si se intenta establecer una demanda y no se encuentra un camino disponible para la misma, se verifica la variable  $d$ , si esta se encuentra en 0 se ejecuta el proceso DS, si no se logra establecer la demanda aun después del proceso de desfragmentación la misma se bloquea y la variable  $d$  se cambia a 1.

La otra posibilidad de ejecución es cuando se intenta liberar una demanda, se ejecuta el proceso DS si  $d = 1$  y si la cantidad de *lighpaths* liberados ( $r$ ) es igual a la variable predefinida previamente  $R$ .

Como se puede ver el método propuesto considera principalmente las conexiones liberadas por lo que puede considerarse como periódica.

Mingyang Zanhg y colegas plantean en su investigación [13] un disparo para el proceso de desfragmentación basado en la cantidad de conexiones liberadas. Básicamente consiste en ejecutar la desfragmentación cuándo la cantidad de conexiones liberadas es igual al parámetro fijo  $TH$  (*Threshold*). En su investigación utilizaron 300 como valor de  $TH$ .

El trabajo presentado por Jie Zhang y colegas [14], proponen un disparo basado en el concepto de *Spectrum Compactness* o Compacidad del Espectro (SC) el cual es una métrica de fragmentación.

Para determinar el momento del disparo para el proceso de desfragmentación tienen en cuenta los siguientes pasos:

- 1) Seleccionar un valor apropiado de *Spectrum Compactness* (SC) para actuar como umbral (T) para el disparo de la desfragmentación.

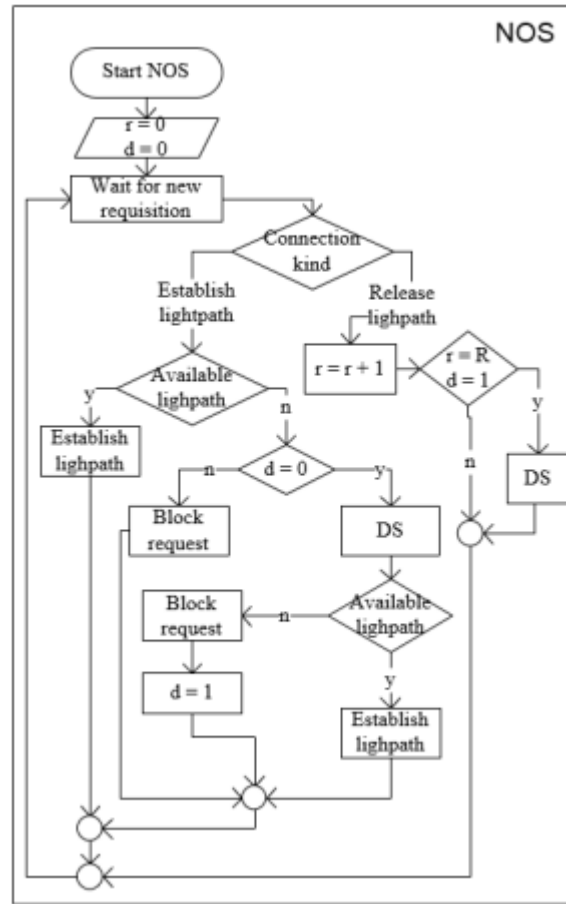


Figura 2.5: Algoritmo propuesto por Favero y colegas [12]

- 2) Actualizar el valor de SC después de liberar conexiones o al establecer una nueva conexión.
- 3) Comparar los valores de SC y T; si  $SC < T$  disparar la desfragmentación y pasar al siguiente paso, sino volver al paso 2.
- 4) Actualizar el ultimo valor de SC después de la desfragmentación y volver al paso 3.

Por último el trabajo propuesto por Zhang y colegas [5] presentan un análisis del problema de la desfragmentación de redes EON dividido en cuatro sub-problemas, el tercero de ellos es ¿Cuándo reconfigurar?.

Para el tercer subproblema plantean un algoritmo de disparo el cual tiene en cuenta la probabilidad de bloqueo instantánea (B) en un periodo  $\Delta t$  ( $B(\Delta t)$ ) y

la utilización del ancho de banda. De esta forma buscan realizar la comparación entre  $B(\Delta t)$  con  $B_{th}$  (umbral de probabilidad de bloqueo para el disparo del proceso de desfragmentación) solo cuando la red se encuentra con un crecimiento en la utilización del ancho de banda.

## Capítulo 3

# Aprendizaje Automático

Aprendizaje Automático o *Machine Learning* es una rama de la inteligencia artificial, el cual construye un modelo matemático basado en datos de muestra, conocidos como datos de entrenamiento, de forma a realizar predicciones o tomar decisiones sin que haya sido programado de forma explícita para realizar dicha tarea.[15]

Una definición más orientada a la ingeniería podría ser:

Se dice que un programa de computadoras aprende de una experiencia  $E$  respecto a una tarea  $T$  y alguna forma de medida de rendimiento  $P$ , si el rendimiento en la tarea  $T$ , medido por  $P$ , mejora con la experiencia  $E$ . [16]

Para entender un poco mejor utilicemos un ejemplo de un programa de aprendizaje de máquinas, como podría ser un filtro de spam de correos electrónicos. Este filtro de spam puede aprender a detectar y marcar los correos mediante ejemplos proveídos por los usuarios, tanto de aquellos correos que el usuario marcó como un spam, así como de correos que el usuario considera que no lo son. Estos datos de ejemplo que el programa puede utilizar para aprender de ellos son llamados conjuntos de entrenamiento.

Para este caso la tarea  $T$  es la de marcar nuevos correos electrónicos como spam, la experiencia  $E$  es el entrenamiento con los datos, y la medida del rendimiento  $P$  podría ser la proporción de acierto o *Accuracy* de los correos correctamente clasificados como spam o no.



### 3.1. Tipos de sistemas de aprendizaje de máquinas

Los sistemas de aprendizaje de máquinas pueden clasificarse de acuerdo a la cantidad y el tipo de supervisión que necesitan durante la fase de entrenamiento. Existen 3 grandes categorías de las cuales hablaremos a continuación:

- **Aprendizaje supervisado:** En este tipo de aprendizaje, el conjunto de entrenamiento que se provee al programa incluye las soluciones deseadas, a estas soluciones se les conoce como etiquetas o *labels*. Este método aprende una regla general la cual mapea el conjunto de entradas con las salidas deseadas. En el aprendizaje supervisado podemos dividir las tareas en dos enfoques principales, la Clasificación y Predicción de un valor numérico generalmente realizado por medio de la Regresión.
  - **Clasificación:** Consiste en brindar ejemplos de entradas indicando a que clase o clases pertenece, de forma a poder realizar el entrenamiento y de esa forma para nuevas entradas poder clasificar dentro de alguna de las clases contempladas.
  - **Predicción:** Consiste en brindar datos de entrada con un conjunto de características y un valor objetivo, de forma que con el entrenamiento para las nuevas entradas es posible predecir el valor objetivo a partir del conjunto de características. Este tipo de tareas se lo conoce como Regresión.
- **Aprendizaje No-supervisado:** En este tipo de aprendizaje, el conjunto de entrenamiento que se provee al programa no incluye las soluciones deseadas o conocidas. La principal tarea de este método es la de encontrar soluciones por si mismo (por ejemplo, patrones o estructuras en los datos).
- **Aprendizaje por refuerzo:** Al tipo de aprendizaje basado en la retroalimentación se lo conoce como aprendizaje por refuerzo. En este método los datos de

entrenamiento son proveídos en forma de recompensas (positivas o negativas) por medio de la retroalimentación a un agente de inteligencia artificial que interactúa con un entorno dinámico. La retroalimentación entre sistema de aprendizaje y la experiencia obtenida por el agente es muy útil para mejorar el rendimiento en la tarea que está aprendiendo.

## 3.2. Redes Neuronales Artificiales

Las redes neuronales artificiales son algoritmos bio-inspirados, es decir, están vagamente inspirados en el cerebro biológico. Como ocurre en el cerebro, las redes neuronales artificiales consisten en unidades simples llamadas neuronas las cuales se encuentran interconectadas entre si. Estas reciben, procesan y transmiten señales a otras neuronas actuando como un interruptor.

Los elementos de una red neuronal son bastantes simples, la complejidad y el poder de estos sistemas proviene de la interacción entre sus elementos. Por ejemplo, el cerebro humano cuenta con más de cien mil millones de neuronas y cien billones de conexiones.

### 3.2.1. Perceptrón

El concepto de perceptrón se encuentra inspirado en las neuronas biológicas, cuya tarea principal es la de bloquear o dejar pasar las señales. Las neuronas reciben estas señales eléctricas, si la señal sobrepasa cierto umbral, la neurona lanza una señal de salida.

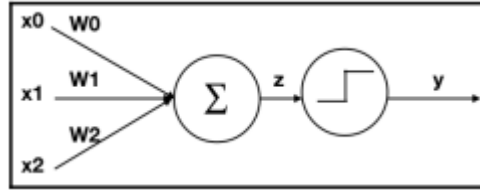


Figura 3.1: Comportamiento del perceptrón

Un perceptrón busca imitar ese mismo comportamiento, de la manera en la que se muestra en la figura 3.1. El mismo puede recibir múltiples datos de entrada  $X = x_0, x_1, x_2, \dots, x_n$ , entonces estos datos son multiplicados por un conjunto de pesos  $W = w_0, w_1, w_2, \dots, w_n$ . La suma ponderada  $z$  de estos datos de entrada, pasa por una función de activación, devolviendo así un valor y dependiendo del tipo de función de activación. También es posible expresarlo matemáticamente por medio de la siguiente fórmula:

$$z = \sum_{i=1}^n w_i x_i = w^T x \quad (3.1)$$

Este es el modelo matemático de una neurona, representado como una suma explícita y como una operación matricial. El término  $w^T x$  es la representación vectorizada de la fórmula, donde  $w$  es el peso de la matriz que primeramente es transpuesta y luego es multiplicada por un vector de entradas  $x$ . Para obtener una descripción matemática completa podríamos añadir una constante  $b$ , llamado bias.

$$z = \sum_{i=1}^n w_i x_i + b = w^T x + b \quad (3.2)$$

Con esto obtenemos una expresión genérica de una ecuación lineal, el cual es todo el proceso previo a su paso por la función de activación, la cual determinara si el perceptrón deja pasar la señal. Existen muchas variantes para la función de activación, La función de activación mas sencilla es la función de paso. La salida de la neurona será aproximada por esta función, la cual se expresa por medio de la siguiente ecuación:

$$f(x) = \begin{cases} 1 & \text{si } (w^T x + b) > 0 \\ 0 & \text{caso contrario} \end{cases} \quad (3.3)$$

### 3.2.2. Funciones de activación

Existen varios tipos de funciones de activación, donde dependiendo de la tarea o valor buscado, estas pueden ser más o menos útiles. A su vez, estas funciones usualmente son utilizadas para agregar la no-linealidad, ya que sin esto solo podríamos tener soluciones lineales. Algunas de las funciones de activación más utilizadas son la sigmoide, tanh y la relu (*rectified linear unit*) los cuales se muestran en la figura 3.2

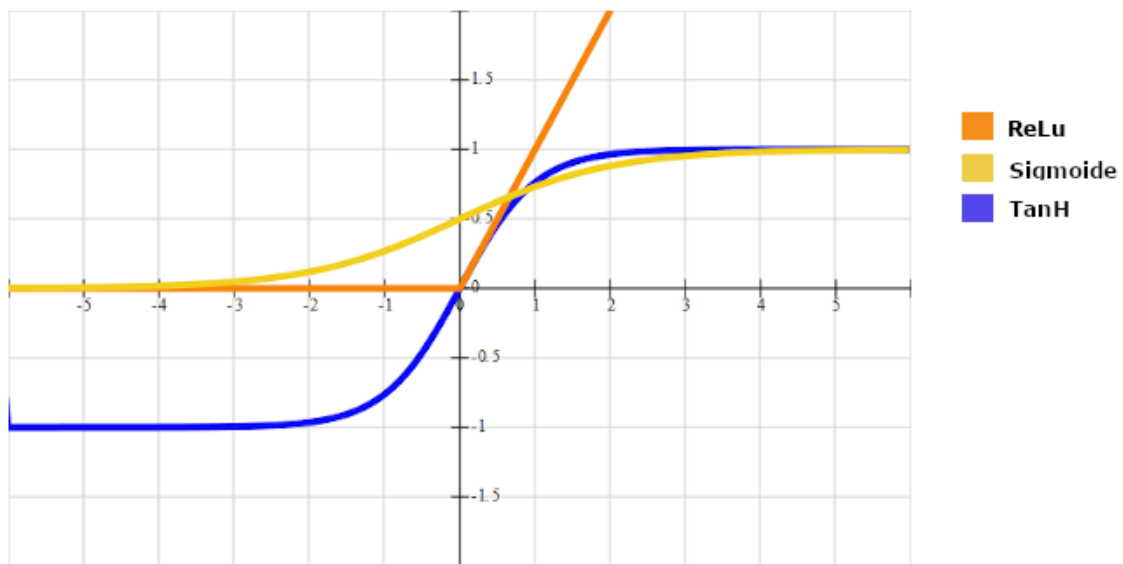


Figura 3.2: Funciones de Activación

### 3.2.3. Arquitectura y aprendizaje

Si muchas neuronas son conectadas entre sí, una red neuronal artificial es creada. En general, toda neurona de la red neuronal artificial está asociada a una capa. Cada neurona está conectada a todas las neuronas de la capa anterior y de la siguiente. Estas conexiones están ponderadas por un peso.

Un ejemplo de una arquitectura de redes neuronales artificiales se puede observar en la figura 3.3, donde se encuentra dividida en tres áreas. La primera área contiene a la capa de entrada, y es la encargada de recibir y alimentar a la red con los datos de entrada. El área intermedia está conformada por una o más capas y son conocidas como capas ocultas, donde el número de capas ocultas determina la profundidad de la red. Por último, la última área contiene a la capa de salida que dependiendo de la tarea puede tener uno o más neuronas de acuerdo a los posibles valores de salida.

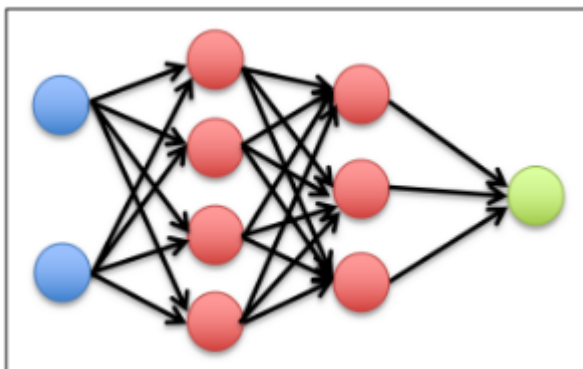


Figura 3.3: Ejemplo de red neuronal

Una vez que el peso de cada una de las conexiones de la red es establecido, es posible calcular un valor de salida para cada entrada de la red neuronal. En la mayoría de los casos, los pesos son establecidos de manera aleatoria por lo que los valores de salida calculados pueden ser diferentes a lo esperado. La idea es ajustar estos pesos de manera individual de forma a que el error cuadrático medio entre los valores calculados y los valores esperados sean mínimos, esto se realiza por medio del método de retropropagación o *backpropagation*.

Primeramente, se calcula el resultado para un registro del conjunto de datos conocido, luego se calcula el error entre el resultado calculado y el resultado esperado. El error es propagado hacia atrás, pasando por todas las capas, empezando por la última capa y terminando con los pesos individuales de la primera capa de la red neuronal. La distribución del error toma lugar como una función de los valores de los pesos individuales. Una vez que los pesos individuales son cambiados, se procede a realizar el mismo procedimiento con un nuevo registro del conjunto de datos.

Para minimizar el error, se utiliza el método de descenso del gradiente, el cual al igual que el algoritmo de retropropagación son explicados en detalle en [17].

Por otro lado tenemos el ratio de aprendizaje o *Learning Rate* el cual es un indicador de cuánto cambian los pesos en cada iteración, Un valor de *Learning rate* alto nos acerca de forma más rápida al error mínimo, pero una valor más bajo puede converger mejor. En la mayoría de los casos los valores elegidos son de 0.001 a 0.3.

Generalmente el entrenamiento de una red neuronal artificial con un solo conjunto de datos no es suficiente para alcanzar el mínimo error de salida, esto especialmente para valores bajos de *Learning Rate*. Para mejorar esto, se realizan varias pasadas del conjunto de datos completo a través de la red neuronal, a cada pasada se lo conoce como época o *epoch*.

### 3.3. Aplicación de Machine Learning en Redes Ópticas Elásticas

En esta sección presentamos un estudio bibliográfico del estado del arte de algunas técnicas de *Machine Learning* aplicadas a problemas existentes en redes ópticas elásticas.

Panchali Datta Choudhury y Tanmay De, nos presentan un análisis del estado del arte del uso de técnicas de *Machine Learning* en redes ópticas elásticas [18].

Algunas de las áreas presentadas por los autores en donde actualmente se utilizan estas técnicas son las siguientes:

- Evaluación o predicción de la calidad de servicio

Investigaciones como la propuesta en [19], presenta un modelo de asignación de ancho de banda en EON teniendo en consideración los requisitos de la calidad del servicio o *Quality of Service* (QoS) de la red. Utilizando la técnica de Aprendizaje Reforzado o *Reinforcement Learning* para resolver el problema de asignación de ancho de banda, donde la función de recompensa para el modelo de aprendizaje reforzado se basa en el cumplimiento o no de los requisitos de QoS.

- Supervivencia

Se hace uso del aprendizaje por refuerzo profundo en el trabajo [20], explorando el problema de optimización de una red en general teniendo en cuenta la capacidad de supervivencia de la misma. Para ello se utilizó un enfoque de aprendizaje por refuerzo profundo, utilizando dos agentes, en donde uno de ellos se utiliza para proporcionar un esquema de trabajo y el otro se encarga de proporcionar el esquema de protección, de esta manera la combinación de ambos sumado a un enfoque de recompensas para aumentar la rentabilidad nos

brinda una solución para las políticas de asignación de espectro, modulación y enrutamiento que además nos provea de supervivencia.

- Predicción de tráfico

El trabajo presentado por Aibin [21] para la predicción de tráfico en redes de centro de datos en la nube o *Cloud Data Center Networks* utiliza un enfoque de árbol de búsqueda de Monte Carlo. Para una solicitud particular, esta técnica de búsqueda identifica el centro de datos en la nube más apropiado y la combinación de rutas candidatas para enrutar las solicitudes. Para lograr esto se crea un árbol disperso realizando la selección por medio del método de Monte Carlo.

- Enrutamiento, modulación y asignación del espectro

En [22], los autores proponen un modelo de aprendizaje reforzado profundo, con el objetivo de desarrollar un modelo autónomo de RMSA en redes ópticas elásticas, utilizando redes neuronales convolucionales también conocidas como *Q Networks* para aprender las políticas RMSA considerando la conectividad, utilización del espectro y solicitudes de tráfico en redes EON.

En el área de interés de este trabajo, nos encontramos con algunos trabajos enfocados a la solución del problema de la fragmentación, pero enfocados en otro tipo de redes, como el presentado en [23], el cual se encuentra enfocado en *Space Division Multiplexing Elastic Optical Networks* o SDM-EON , utilizando redes neuronales, específicamente una conocida como red neuronal de Elman, para la predicción de tráfico de forma a mitigar la fragmentación y el problema de *Cross-talk*.

Para este tipo de redes también contamos con un algoritmo de desfragmentación basado en Machine Learning [24]. Los autores proponen un algoritmo de desfragmentación utilizando un enfoque de aprendizaje no supervisado, por lo que no requiere de conocimientos previos de la red. El algoritmo se encarga de identificar aquellos



*lightpaths* que pueden ser agrupados en base a ciertas características, para luego mapear esos grupos a los núcleos y reordenar al espectro sin necesidad de realizar re-ruteos.

# Capítulo 4

## Método Propuesto

Este trabajo propone una solución para la selección del momento de disparo de procesos proactivos de desfragmentación en redes EON, basado en técnicas de *Machine Learning* para los cuáles predicen la probabilidad de bloqueo de una demanda *unicast* en un momento determinado  $t$ .

Para esta predicción se utiliza un modelo de regresión basado en redes neuronales, utilizando un conjunto de datos de simulaciones de tráfico *unicast* en diversas topologías de redes EON, tomando parámetros o características relacionadas a la fragmentación y la utilización de la red como datos de entrada y produciendo un valor estimado de la probabilidad de bloqueo. Se fija además un valor límite de probabilidad para la realización del proceso de desfragmentación.

### 4.1. Características

En el área de *Machine Learning*, se conoce cómo “características” a los parámetros o datos de entrada del modelo de aprendizaje.

Las características seleccionadas fueron aquellas que se encuentran relacionadas al uso y la fragmentación de la red, así como también al bloqueo de las demandas. Se tomaron las principales métricas usadas para la determinación del estado de fragmentación, de acuerdo con [25], además de otras relacionadas a la utilización de

la red.

Estas características son las siguientes:

- Entropía de utilización[26]: La entropía de utilización es una métrica de fragmentación de enlaces de la red. Un valor bajo de entropía indica que el ancho de banda de los enlaces de fibra óptica está siendo usado de forma ordenada, con menos bloques de FS utilizados o no utilizados, y con un nivel de fragmentación menor. La entropía de un enlace está definida como:

$$Ent_{link} = \sum_{i=1}^{N-1} FS_i \oplus FS_{i+1} \quad (4.1)$$

donde N es la cantidad de FS en el enlace,  $FS_i$  representa al FS de índice i dentro del enlace, y tiene valor 1 si el FS está ocupado, y valor 0 en caso contrario, y se realiza una operación XOR sobre FS contiguos del enlace. Para obtener la entropía de la red se calcula el promedio de la entropía en cada enlace:

$$Ent_{red} = \sum_{i=1}^{|E|} \frac{Ent_{link-i}}{|E|} \quad (4.2)$$

- Entropía de Shannon (SHF)[27]: Es una métrica de fragmentación de enlaces, que es una variación de la anterior característica, definida por.

$$SHF_{link} = \sum_{i=1}^B \frac{S_i^{free}}{N} \ln \frac{N}{S_i^{free}} \quad (4.3)$$

Donde  $S^{free}$  representa la cantidad de FS libres en el enlace y B la cantidad de bloques de FS libres. Para calcular el SHF de la red se calcula el promedio de los valores calculados en todos los enlaces.

$$SHF_{red} = \sum_{i=1}^{|E|} \frac{SHF_{link-i}}{|E|} \quad (4.4)$$

- *Bandwidth Fragmentation Ratio* o Relación de Fragmentación de ancho de banda o (BFR)[13]: Representa el índice de fragmentación de los recursos de la red. El BFR de un link se define como:

$$BFR_{link} = 1 - \frac{MaxBlock()}{S^{free}} \quad (4.5)$$

Donde  $MaxBlock()$  es el tamaño del mayor bloque de FS bloqueados y  $S^{free}$  es la sumatoria total de FS libres en el enlace  $link$ . El BFR de la red podemos calcular de la siguiente manera.

$$BFR_{red} = \sum_{i=1}^{|E|} \frac{BFR_{link-i}}{|E|} \quad (4.6)$$

- *Maximum Slot Index* o Mayor Índice de FS utilizado (MSI): Este valor indica el valor del índice de FS más alto que está siendo utilizado dentro de un enlace. Para calcular el MSI de la red, se halla el índice máximo usado en todos los enlaces de la red y se calcula el promedio:

$$MSI_{red} = \sum_{i=1}^{|E|} \frac{MSI_{link-i}}{|E|} \quad (4.7)$$

Donde  $MSI_{link-i}$  es el mayor índice utilizado en el enlace  $i$ .

- Consecutividad del espectro (CE)[28]: Este valor refleja la alineación de los FS en un camino en particular, para obtener el valor para un camino en particular aplicamos la siguiente fórmula.

$$CE = \frac{Joins}{Bloques} \times \frac{S_{free}}{N} \quad (4.8)$$

Donde  $Joins$  se calcula como la cantidad total de bloques de dos FS libres adyacentes distintos dentro del enlace,  $Bloques$  es la cantidad de bloques de FS libres en el enlace y  $S_{free}$  es la cantidad de FS libres en el enlace. Y para calcular la consecutividad en una red se calculan previamente todos los caminos de dos enlaces en la red, y luego se calcula el valor de la consecutividad para todas esas rutas y al final se halla el promedio.

$$CE_{red} = \sum_{i=1}^{|K|} \frac{CE_{link-i}}{|K|} \quad (4.9)$$

donde  $CE_{link-i}$  representa la consecutividad de la ruta de dos enlaces  $i$  calculada previamente, y  $K$  es la cantidad de rutas de dos enlaces que existen en la red.

- Utilización de la Red: Se define como el cociente entre la sumatoria de todas las FS ocupados con la cantidad total de FS.

$$U_{so} = \sum_{i=1}^{|E|} \frac{sum(i)}{N} \quad (4.10)$$

Donde  $sum(i)$  es la cantidad de FS utilizados en el enlace  $i$  y  $N$  es la cantidad total de FS en el enlace.

- Acumulación de FS bloqueados: Valor que muestra la sumatoria de las FS requeridos por demandas bloqueadas en las  $D$  demandas anteriores al periodo de tiempo actual.

$$FSB = \sum_{i=0}^D S_i^{block} \quad (4.11)$$

Donde  $S_i^{block}$  es la cantidad de FS solicitados por la demanda bloqueada  $i$ .

En la figura 4.1 se puede ver un ejemplo de una topología con 4 conexiones activas, el estado de sus enlaces y el cálculo de cada una de las métricas de fragmentación explicadas anteriormente.

## 4.2. Obtención de datos para el entrenamiento

Se utilizó un simulador de redes EON [9] para la generación del conjunto de datos a ser utilizados para el entrenamiento de la red neuronal.

Para esto se utilizaron dos topologías de red: USNET y NSFNET, en donde por cada topología se realizaron 50 simulaciones con volumen de tráfico variable en la misma simulación, para lograr que la cantidad de conexiones activas no permanezca constante durante la simulación. La variación de tráfico usada fue la que se ve en la figura 4.2-a, el proceso se realizó durante 1210 unidades de tiempo para cada simulación, generando un total de 121.000 registros.

Una vez generados los datos, los mismos fueron preprocesados, de forma a obtener los valores de la probabilidad de bloqueo que deseamos estimar.

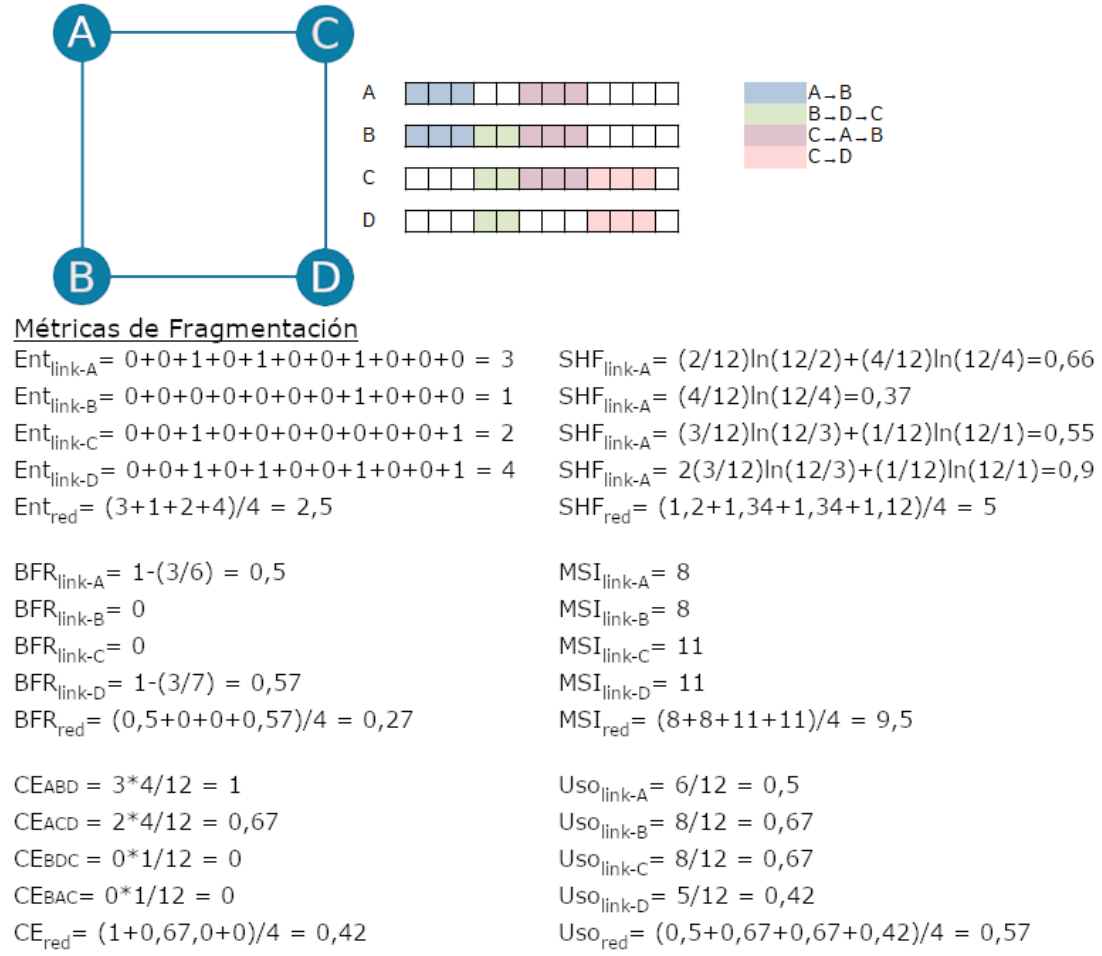


Figura 4.1: Ejemplo de métricas de fragmentación

El preprocesamiento consiste en el cálculo del índice de bloqueo en base a la ecuación 4.12, donde para un tiempo  $t$ , utilizando una ventana de 10 unidades de tiempo hacia delante, para cada instante podemos obtener una mirada hacia delante de posibles bloqueos,  $FSB_i$  es la sumatoria de  $FS$  bloqueados en el tiempo  $i$  y  $FSD_i$  es la sumatoria de  $FS$  demandados en el tiempo  $i$ .

$$PB_t = \frac{\sum_{i=t}^{t+T} FSB_i}{\sum_{i=t}^{t+T} FSD_i} \quad (4.12)$$

Además, una vez separados los datos y debido a la diferencia de rangos de valores, los mismos son normalizados, usando la siguiente fórmula.

$$n = x - \frac{train_{mean}}{train_{std}} \quad (4.13)$$

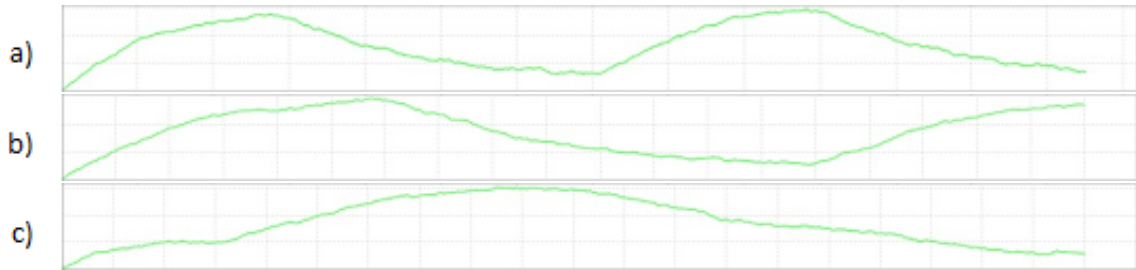


Figura 4.2: Volumen de tráfico variado utilizado

Donde  $x$  es el valor que queremos normalizar,  $train_{mean}$  la media de valores y  $train_{std}$  es la desviación estándar.

### 4.3. Herramientas Utilizadas

En esta sección se presentan las herramientas utilizadas para la implementación del método propuesto en este trabajo, seleccionadas luego de realizar numerosas ejecuciones de prueba para obtener los valores de los parámetros del modelo, así como la prueba de concepto en sí.

Para todo el proceso de *Machine Learning* se utilizó la plataforma de código abierto llamada **TensorFlow** [29] en el lenguaje de programación **Python**. En la creación y entrenamiento de redes neuronales se utilizó la API de alto nivel incluida en la plataforma **Tensorflow** denominada **Keras**, la cual permite la creación de modelos de aprendizaje automático de forma rápida y sencilla.

Elegimos Tensorflow como herramienta principal debido a que es una plataforma de código abierto orientado a *Machine Learning*. Cuenta con un ecosistema completo de herramientas y librerías que facilitan la creación y desarrollo de aplicaciones de aprendizaje automático, además cuenta con una extensa y muy completa documentación.

## 4.4. Modelado

Para entrenar los datos recolectados se utilizó un modelo con una capa de entrada de 7 neuronas, dos capas ocultas densamente conectadas, cada una con 64 y 32 neuronas respectivamente, y una capa de salida que devuelve un único valor continuo. Para todas las capas la función de activación utilizada fue la RELU (Ver sección 3.2.2).

## 4.5. Entrenamiento

Para el entrenamiento del modelo, se creó un conjunto de datos de entrenamiento utilizando el 70 % de los datos recolectados de forma aleatoria.

El 20 % de los datos de entrenamiento se utilizó como el conjunto de validación. Una técnica utilizada en el procedimiento es el llamado parada temprana o *Early Stopping*, el cual mediante el monitoreo del rendimiento del entrenamiento nos permite detenernos una vez que el error de validación aumente de forma sostenida, de forma así evitar un sobre entrenamiento.

El modelo se entrena como máximo por 1000 épocas, el cual se detiene automáticamente cuando el valor del error de validación deja de mejorar. La figura 4.3 muestra la evolución del error al pasar las épocas.

Los parámetros comparados son el error de entrenamiento o *Train Error*, el cuál es el error obtenido durante la fase de entrenamiento del modelo y el error de validación o *Val Error*, que se obtiene en la fase de validación.

Por cada época el *Val Error* es comparado con el *Train Error*, esto hasta que se determina que ya no existe mejora, sino que el error de validación se mantiene o aumenta su valor con respecto al punto de parada.



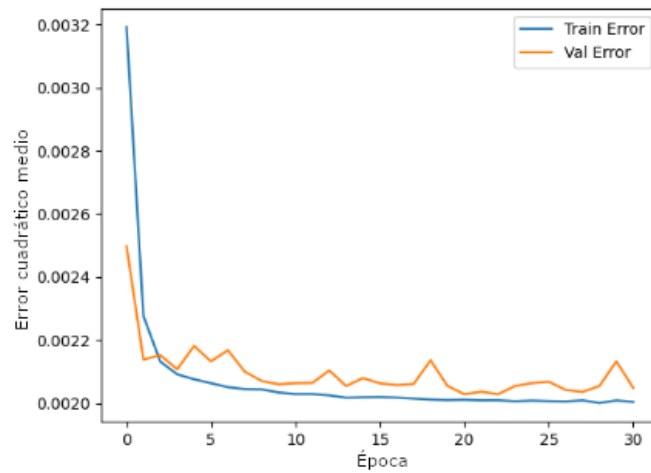


Figura 4.3: Evolución del error a través de épocas

## 4.6. Pruebas de predicción

Para comprobar la efectividad de nuestro modelo se procedió a tomar el 30 % de datos restantes que no fueron incluidos en el entrenamiento y realizar predicciones, como ya conocemos el valor de la probabilidad de bloqueo que se busca predecir podemos calcular el error absoluto medio (MAE) y el error cuadrático medio (MSE). La tabla 4.1 muestra los resultados obtenidos, siendo éstos muy satisfactorios.

Tabla 4.1: Tabla de resultados en pruebas de predicción

MAE	0.0239
MSE	0.002

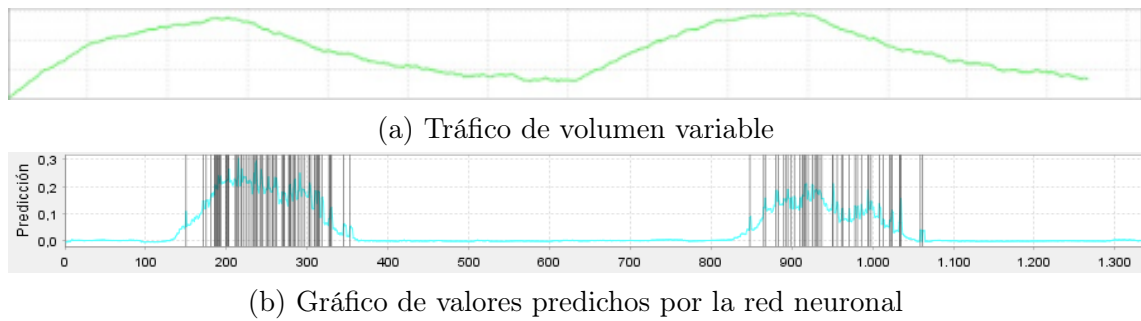


Figura 4.4: Utilización de la red y predicciones

En la Figura 4.4 - a se observa la variación de la utilización de la red, al realizarse una simulación con volumen de tráfico variable sin realizar procesos de desfragmentación. En la parte b de la misma figura se observa la curva de valores predichos con las técnicas presentadas junto a los bloqueos dados en la misma simulación. Puede observarse cómo este valor acompaña la variación de la utilización de la red, y también toma valores más altos en periodos de tiempo en que la frecuencia de bloqueos se hace más alta.

## Capítulo 5

# Pruebas y resultados obtenidos

Las pruebas se realizaron sobre el simulador de redes elásticas ópticas desarrollado en [9] con las topologías USNET y NSFNET, la generación de demandas se realiza de manera aleatoria, pero con variaciones de volumen de tráfico. En la figura 4.2 se pueden ver las curvas de rutas activas, siendo el eje  $x$  el tiempo y el eje  $y$  la cantidad de rutas activas, para lograr esto inicialmente la simulación inicia con 700 Erlangs y al llegar la utilización de la red a un valor máximo baja a 100 Erlangs, de esta forma conseguimos que la carga de la red no sea constante y podamos acercarnos a un escenario más real. Para el proceso de desfragmentación se utilizó el algoritmo genético propuesto en [9].

El proceso que seguimos para seleccionar el periodo de tiempo en el que el proceso de desfragmentación se ejecutará se puede ver en la figura 5.1, donde las características son las citadas en el capítulo 4 sección 4.1,  $PB$  es la probabilidad de bloqueo calculada por nuestro modelo entrenado,  $PB_{th}$  es el valor seleccionado como umbral para que el proceso de desfragmentación se dispare y  $DF$  es el proceso de desfragmentación.

Para tener un punto de referencia nuestro modelo se compara con otros dos métodos conocidos:

- a) Desfragmentación periódica por tiempo fijo: el cual consiste en desfragmentar la red cada cierta cantidad constante de unidades de tiempo.

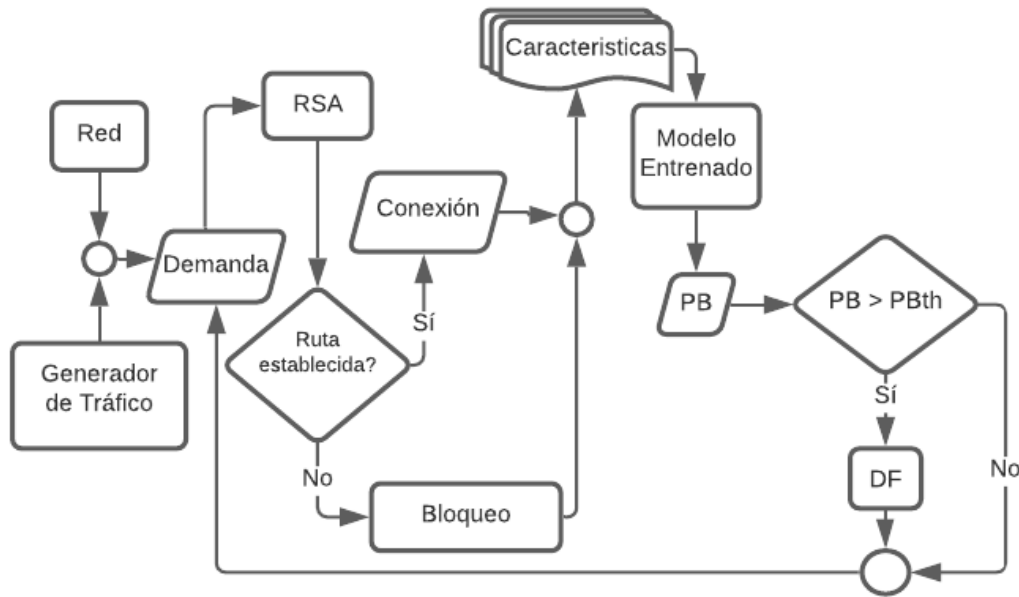


Figura 5.1: Diagrama simulador / modelo entrenado

- a) Desfragmentación por métricas de fragmentación: el cual consiste en desfragmentar cuando la red alcanza cierto valor de alguna métrica, en este caso de BFR.

Cada método tiene su propio parámetro encargado de disparar el proceso de desfragmentación y durante las pruebas se hace variar este parámetro hasta tres veces.

Para elegir este valor tomamos como base la cantidad de desfragmentaciones que realiza el método por tiempo fijo y se buscan valores para los parámetros de los otros métodos de tal manera que ejecuten cantidades similares de desfragmentaciones, así logramos que la comparación se haga de la forma más justa posible.

Las simulaciones se realizaron con los tres métodos bajo las mismas condiciones de variaciones de distribuciones demandas y parámetros, para obtener mayor variación en los resultados este procedimiento se repitió 5 veces.

## 5.1. Objetivos a optimizar

Para la evaluación de los resultados, considerando dos objetivos globales, medidos al final de cada simulación:

- Obj. 1) **Cantidad de bloqueos(BL)**: Suma de las demandas bloqueadas durante la simulación
- Obj. 2) **Cantidad de reconfiguraciones(RC)**: Número de conexiones reconfiguradas durante los procesos de desfragmentación realizadas durante la simulación.

Como se trata de dos objetivos a optimizar (BL y RC), buscando minimizar el valor de ambas, para la comparación de los resultados se consideraron dos métricas de desempeño para optimización multi-objetivo [30]:

- **Número de soluciones en el Frente Pareto obtenido (SFP)**: Para esta primera métrica, se busca determinar el método que contenga el mayor número de soluciones no dominadas que minimicen BL y RC, comparando al mismo tiempo las soluciones de nuestro método propuesto con las soluciones de los otros dos métodos.
- **Cobertura Pareto(CP)**: Como segundo método de comparación utilizamos una métrica binaria denominada “Cobertura”, la cual nos permite realizar la comparación de las soluciones óptimas de nuestro método propuesto con los demás métodos tomándolos de a pares.

$$C(A, B) = \frac{\text{Soluciones de } B \text{ dominadas por } A}{\text{Soluciones totales de } B} \quad (5.1)$$

## 5.2. Análisis de los resultados

Se puede observar en la Tabla 5.1 que de todas las instancias experimentales que se encuentran en el Frente Pareto, el método propuesto (MP) en este

trabajo representa el 52.9% de los puntos dentro del mismo, lo que indica que este método produce más soluciones no dominadas, indicando una mayor eficiencia en lo que respecta a la minimización de la cantidad de bloqueos y el número de reconfiguraciones en la red.

Tabla 5.1: Tabla de soluciones en el frente pareto

SIMULACIÓN	Soluciones del Frente Pareto			
	MP	Tiempo Fijo	BFR	Total
NSFNET	<b>9</b>	4	6	19
USNET	<b>9</b>	4	2	15

En las figuras 5.2 y 5.3 podemos observar los resultados para los tres diferentes tipos de tráfico, para el caso de la topología USNET (figura 5.3), en el frente de soluciones óptimas se aprecia una predominancia del método propuesto en cuanto a cantidad de soluciones. Mientras que en el caso de la topología NSFNET (figura 5.2), dicha predominancia se puede ver en dos de los tres tipos de tráfico.

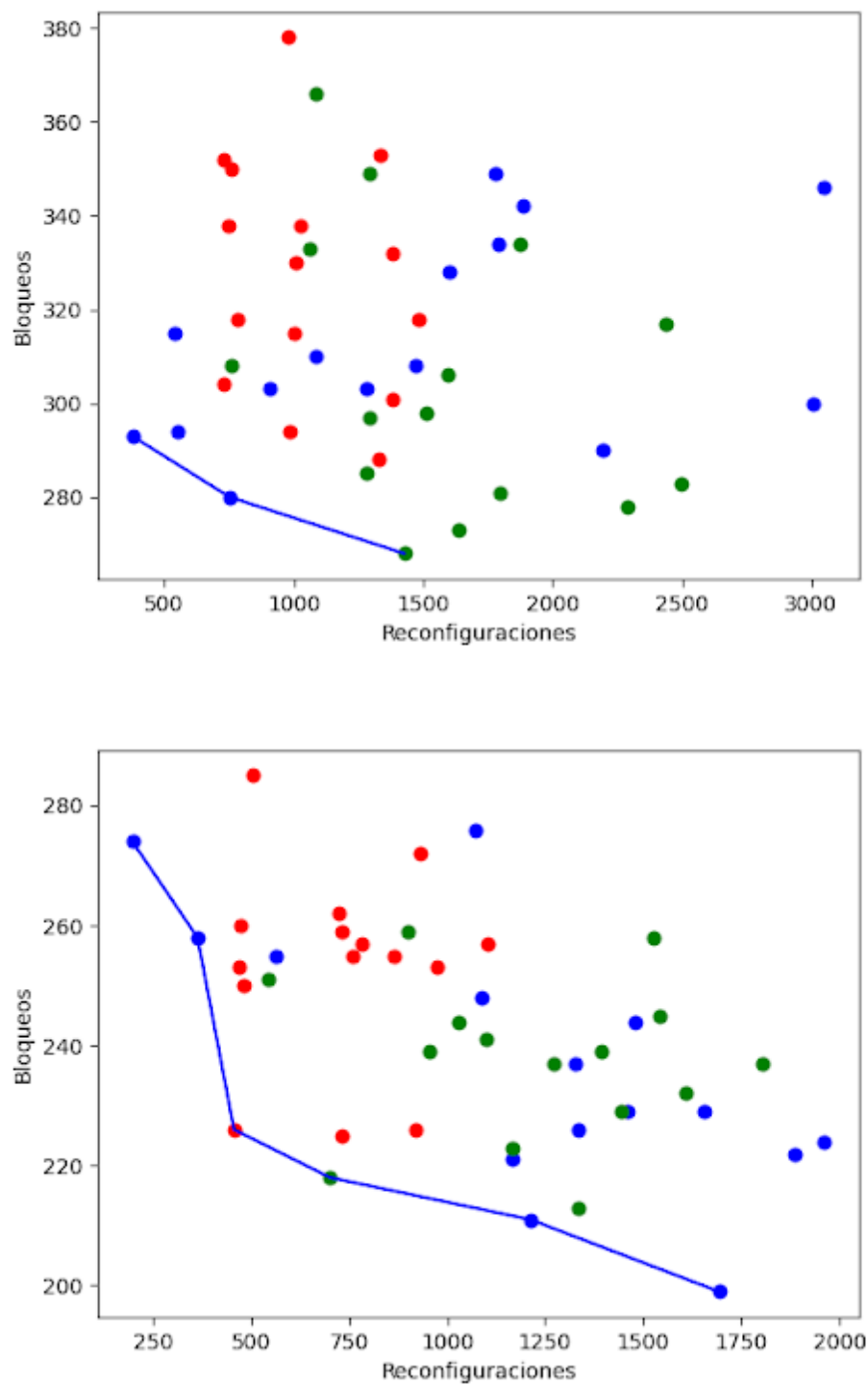


Figura 5.2: Gráfico de soluciones en el frente pareto para la topología NSFNET

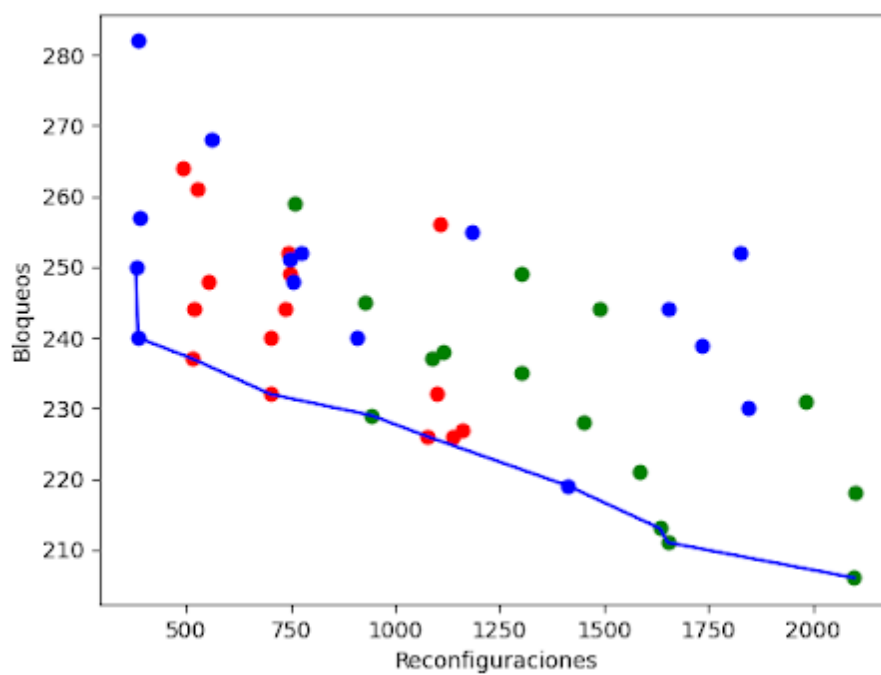


Figura 5.2: Gráfico de soluciones en el frente pareto para la topología NSFNET

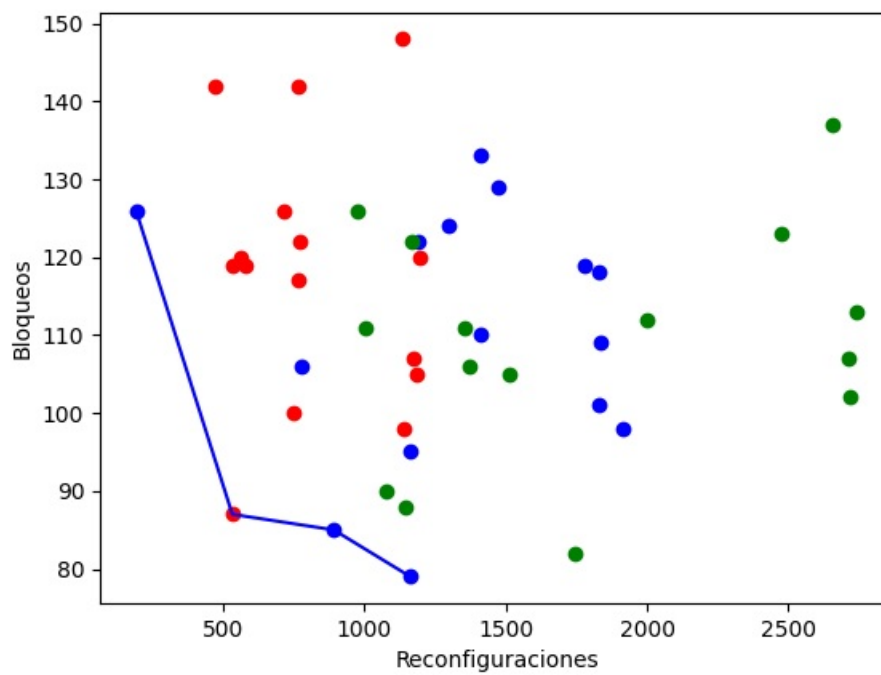


Figura 5.3: Gráfico de soluciones en el frente pareto para la topología USNET



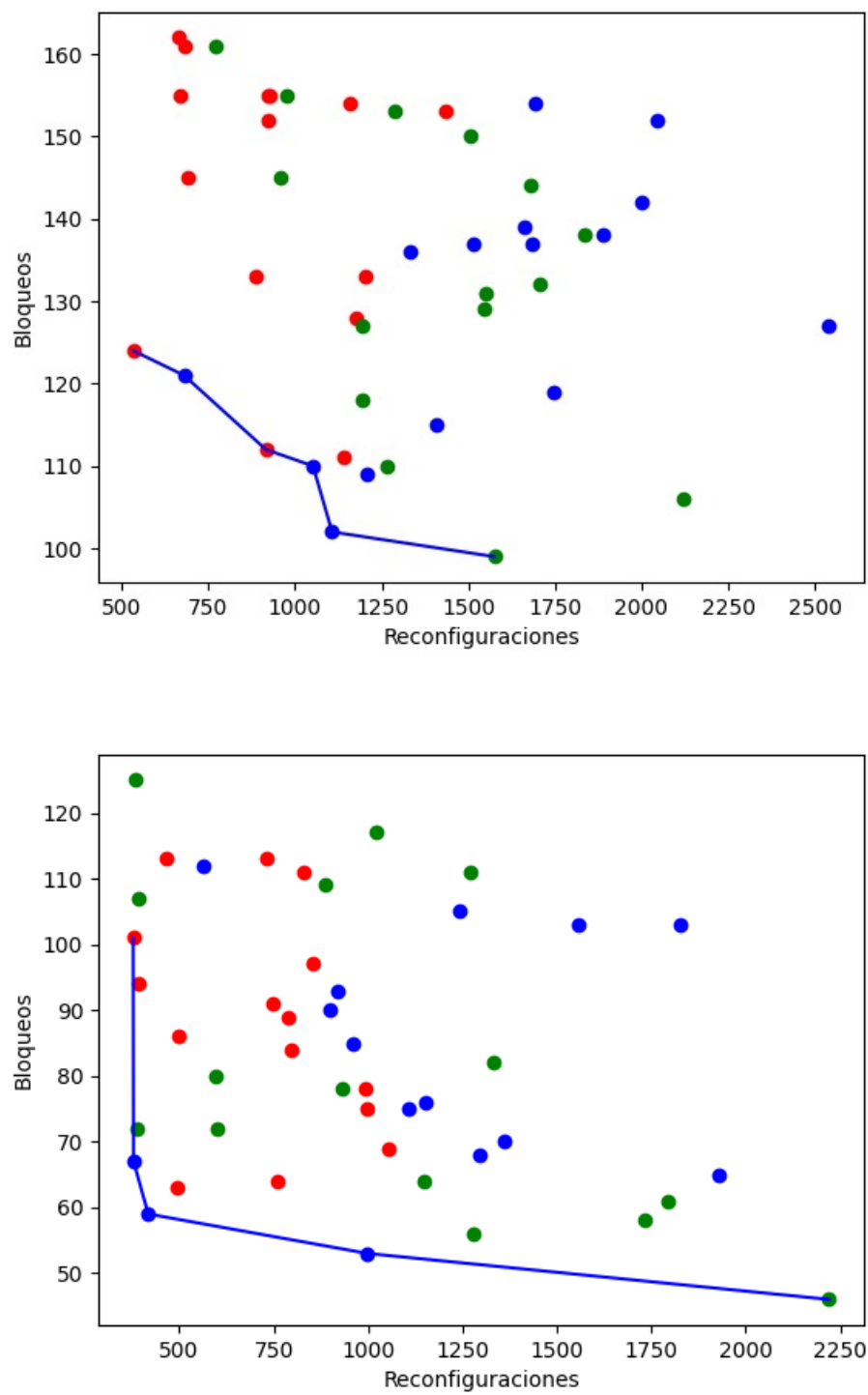


Figura 5.3: Gráfico de soluciones en el frente pareto para la topología USNET

En las tablas 5.2 y 5.3 se presentan los resultados de la métrica CP, para los tres tipos de tráfico, donde *MP* es el método propuesto, *TF* el método por tiempos fijos y *BFR* es el método por métricas. Se verifica que en un 75 % del total de comparaciones, incluidas las dos topologías, el método propuesto obtiene mayor cobertura en las soluciones óptimas.

Tabla 5.2: Tabla de cobertura para topología USNET

USNET					
Tráfico	A	B	C(A,B)	C(B,A)	Conclusión
4.2-a	MP	TF	0,5	0,25	<b>A cubre a B en un 50 % y es cubierto en un 25 %</b>
4.2-a	MP	BFR	1	0	<b>A cubre a B en un 100 % y es cubierto en un 0 %</b>
4.2-b	MP	TF	1	0	<b>A cubre a B en un 100 % y es cubierto en un 0 %</b>
4.2-b	MP	BFR	1	0	<b>A cubre a B en un 100 % y es cubierto en un 0 %</b>
4.2-c	MP	TF	0,5	0,67	B cubre a A en un 67 % y es cubierto en un 50 %
4.2-c	MP	BFR	0,33	0	<b>A cubre a B en un 33 % y es cubierto en un 0 %</b>

Tabla 5.3: Tabla de cobertura para topología NSFNET

NSFNET					
Tráfico	A	B	C(A,B)	C(B,A)	Conclusión
4.2-a	MP	TF	0,25	0	<b>A cubre a B en un 25 % y es cubierto en un 0 %</b>
4.2-a	MP	BFR	0,43	0	<b>A cubre a B en un 43 % y es cubierto en un 0 %</b>
4.2-b	MP	TF	0	0,29	B cubre a A en un 29 % y es cubierto en un 0 %
4.2-b	MP	BFR	0,33	0,43	B cubre a A en un 43 % y es cubierto en un 33 %
4.2-c	MP	TF	1	0	<b>A cubre a B en un 100 % y es cubierto en un 0 %</b>
4.2-c	MP	BFR	0,67	0	<b>A cubre a B en un 67 % y es cubierto en un 0 %</b>

## Capítulo 6

# Conclusiones y Trabajos Futuros

En las redes ópticas elásticas, la constante asignación y liberación de recursos en forma dinámica puede dar lugar al problema conocido como "fragmentación del ancho de banda". Este problema es crítico ya que la presencia de bloques aislados de ancho de banda dentro del dominio del espectro deja a los mismos inutilizables ante futuras solicitudes de conexiones, debido a que los mismos no se encuentran alineados y contiguos,

Un enfoque utilizado para combatir la fragmentación son los procesos de desfragmentación, que consisten en el retiro y posterior re-ruteo de un sub-conjunto de conexiones existentes, con el objetivo de consolidar los espacios disponibles en grandes bloques contiguos y continuos que puedan ser utilizados para futuras solicitudes de conexiones.

El problema analizado en este trabajo es el de ¿Cuándo Reconfigurar?, es decir, buscar el momento adecuado para disparar el proceso de desfragmentación, ya que desfragmentaciones muy frecuentes o muy distantes en el tiempo pueden hacer que estos no sean muy eficientes.

Este trabajo presenta un enfoque con desfragmentación para tráfico dinámico de redes ópticas elásticas por medio de un disparador inteligente utilizando técnicas de *Machine Learning*. En su implementación se utilizó un enfoque de aprendizaje supervisado, con un modelo de redes neuronales artificiales para la predicción de futuros

bloqueos y utilizando algunas características para medir el estado de fragmentación de la red y el uso de la misma.

El método propuesto de disparo recibe el estado actual de la red o “características” para cada instante de tiempo, con estas características y el entrenamiento previo del modelo, obtenemos una predicción de la probabilidad de futuros bloqueos, para una ventana de 10 unidades de tiempo hacia delante.

Para evaluar la eficiencia del método propuesto de disparo se consideraron tres escenarios diferentes, con un volumen de tráfico variable, utilizando las topologías NSFNET y USNET. Los objetivos a optimizar fueron:

- La cantidad de bloqueos obtenidos (BL)
- La cantidad de reconfiguraciones al final de cada instancia de prueba (RC)

## 6.1. Conclusiones Experimentales

Se realizaron pruebas experimentales a fin de comparar nuestro método de disparo contra otros dos presentes en la literatura científica. El método de desfragmentación periódica es una estrategia ampliamente utilizada, la cual consiste en realizar el proceso de desfragmentación cada cierto periodo fijo de tiempo y el disparo por medio de métricas, la cual considera en realizar el proceso de desfragmentación en base al valor actual de la métrica, para las pruebas de este método se utilizó la métrica de fragmentación BFR.

Para comparar los resultados obtenidos en relación a los objetivos citados anteriormente (BL y RC), se utilizaron dos métricas de desempeño para optimización multi-objetivo.

1. Número de soluciones en el Frente Pareto (SFP).
2. Cobertura Pareto (CP).

Como resultado de la comparación de los métodos en base a los objetivos expuestos previamente, se concluye que el método propuesto es mejor ya que consigue en la mayoría de los escenarios mejores resultados, minimizando los valores obtenidos para BL y RC. Considerando la métrica SFP se obtiene que constituye el 52.9 % de soluciones no dominadas y en el caso de CP se logró en un 75 % del total de comparaciones resultados favorables a nuestro método.

## 6.2. Aportes

Los aportes del presente trabajo son:

- 1) Un análisis bibliográfico sobre el problema relacionado al periodo de tiempo en el que el proceso de desfragmentación será ejecutado.
- 2) Una investigación y recopilación de métricas de fragmentación las cuales son utilizadas como características necesarias para la predicción de la probabilidad de bloqueo por parte del modelo entrenado.
- 3) Como principal aporte se diseñó un algoritmo que realiza el preprocesamiento de datos, entrenamiento del modelo y predicción de probabilidades de bloqueo utilizando técnicas de *Machine Learning*.
- 4) Pruebas experimentales utilizando en conjunto un simulador de redes EON y el modelo entrenado a fin de evaluar la eficiencia de nuestro método propuesto. Se realizaron comparaciones contra otros dos mecanismos de disparo, disparo periódico en tiempos fijos y disparo basado en la métrica BFR, teniendo resultados favorables para nuestro método en relación a la minimización de la cantidad de bloqueos y reconfiguraciones.

### 6.3. Trabajos Futuros

- Aplicar el modelo de disparo inteligente del proceso de desfragmentación propuesto en este trabajo a redes ópticas elásticas que utilizan otras tecnologías o técnicas, como las redes EON con multiplexación por división de espacios o *Space División Multiplexing* (SDM).

EL SDM es utilizado en redes con múltiples núcleos, por lo que sería interesante realizar un análisis de la eficiencia del modelo en este tipo de redes.

- Proponer algoritmos de disparo del proceso de desfragmentación utilizando métodos estadísticos, tal como la regresión logística binaria (RLB), la cual se utiliza cuando se desea conocer la relación entre una variable dependiente binaria y una o más variables independientes o explicativas, las cuales pueden ser cuantitativas y/o cualitativas.

El objetivo de la RLB es obtener una estimación ajustada de la probabilidad de ocurrencia de un evento a partir de una o más variables independientes.

- Otro enfoque posible es utilizar programación genética, el cual consiste en una metodología basada en algoritmos evolutivos e inspirada en la evolución biológica para desarrollar programas que realicen ciertas tareas, por ejemplo, realizar disparo del proceso de desfragmentación en el mejor momento.

Es una técnica de aprendizaje automático utilizada para optimizar una población de programas de acuerdo a una función de ajuste o *fitness function* que evalúa la capacidad de cada programa de realizar la tarea.

# Bibliografía

- [1] Michal Aibin y Krzysztof Walkowiak. «Defragmentation algorithm for joint dynamic and static routing problems in elastic optical networks with unicast and anycast traffic». En: *2016 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2016, págs. 1-5.
- [2] Masahiko Jinno y col. «Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies». En: *IEEE communications magazine* 47.11 (2009), págs. 66-73.
- [3] Bijoy Chand Chatterjee, Nityananda Sarma y Eiji Oki. «Routing and spectrum allocation in elastic optical networks: A tutorial». En: *IEEE Communications Surveys & Tutorials* 17.3 (2015), págs. 1776-1800.
- [4] Weiran Shi y col. «On the effect of bandwidth fragmentation on blocking probability in elastic optical networks». En: *IEEE transactions on communications* 61.7 (2013), págs. 2970-2978.
- [5] Mingyang Zhang y col. «Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic». En: *Journal of Lightwave Technology* 32.5 (2014), págs. 1014-1023.
- [6] Bijoy Chand Chatterjee, Seydou Ba y Eiji Oki. «Fragmentation problems and management approaches in elastic optical networks: A survey». En: *IEEE Communications Surveys & Tutorials* 20.1 (2017), págs. 183-210.

- [7] Sahar Talebi y col. «Spectrum management techniques for elastic optical networks: A survey». En: *Optical Switching and Networking* 13 (2014), págs. 34-48.
- [8] Jaume Comellas, Laura Vicario y Gabriel Junyent. «Periodic Defragmentation in Elastic Optical Networks». En: *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE. 2018, págs. 1-4.
- [9] Enrique J Dávalos y col. «Spectrum defragmentation in elastic optical networks: Two Approaches With Metaheuristics». En: *IEEE Access* 7 (2019), págs. 119835-119843.
- [10] Jie Luo y col. «Partial defragmentation in flexible grid optical networks». En: *Asia Communications and Photonics Conference*. Optical Society of America. 2012, AF4A-54.
- [11] Yutaka Takita y col. «Wavelength defragmentation with minimum optical path disruptions for seamless service migration». En: *Optical Fiber Communication Conference*. Optical Society of America. 2016, M2J-3.
- [12] Ricardo V Fávero y col. «A new elastic optical network defragmentation strategy based on the reallocation of lightpaths sharing the most fragmented link». En: *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*. IEEE. 2015, págs. 1-5.
- [13] Mingyang Zhang y col. «Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions». En: *2013 IEEE International Conference on Communications (ICC)*. IEEE. 2013, págs. 3894-3898.
- [14] Jie Zhang y col. «Priority-based defragmentation scheme in spectrum-efficient optical networks». En: *The 10th International Conference on Optical Internet (COIN2012)*. IEEE. 2012, págs. 16-17.
- [15] Xian-Da Zhang. *A matrix algebra approach to artificial intelligence*. Springer, 2020, págs. 223-440.



- [16] Tom Mitchell. «Machine learning». En: (1997).
- [17] Tariq Rashid. *Make your own neural network*. CreateSpace Independent Publishing Platform, 2016.
- [18] Panchali Datta Choudhury y Tanmay De. «Recent developments in Elastic Optical Networks Using Machine Learning». En: *2019 21st International Conference on Transparent Optical Networks (ICTON)*. IEEE. 2019, págs. 1-3.
- [19] Tania Panayiotou y col. «A data-driven bandwidth allocation framework with QoS considerations for EONs». En: *Journal of Lightwave Technology* 37.9 (2019), págs. 1853-1864.
- [20] Xiao Luo y col. «Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability». En: *Optics express* 27.6 (2019), págs. 7896-7911.
- [21] Michal Aibin. «Traffic prediction based on machine learning for elastic optical networks». En: *Optical Switching and Networking* 30 (2018), págs. 33-39.
- [22] Xiaoliang Chen y col. «Deep-RMSA: A deep-reinforcement-learning routing, modulation and spectrum assignment agent for elastic optical networks». En: *2018 Optical Fiber Communications Conference and Exposition (OFC)*. IEEE. 2018, págs. 1-3.
- [23] Silvana Trindade y Nelson LS da Fonseca. «Machine Learning for Spectrum Defragmentation in Space-Division Multiplexing Elastic Optical Networks». En: *IEEE Network* 35.1 (2020), págs. 326-332.
- [24] Yu Xiong y col. «A machine learning approach to mitigating fragmentation and crosstalk in space division multiplexing elastic optical networks». En: *Optical Fiber Technology* 50 (2019), págs. 99-107.
- [25] M Quagliotti y col. «Spectrum fragmentation metrics and their use in optical channel allocation algorithms». En: (2017).

- [26] Xi Wang y col. «Utilization entropy for assessing resource fragmentation in optical networks». En: *Optical Fiber Communication Conference*. Optical Society of America. 2012, OTh1A-2.
- [27] Paul Wright, Michael C Parker y Andrew Lord. «Minimum-and maximum-entropy routing and spectrum assignment for flexgrid elastic optical networking». En: *Journal of Optical Communications and Networking* 7.1 (2015), A66-A72.
- [28] Ying Wang y col. «Spectrum consecutiveness based routing and spectrum allocation in flexible bandwidth networks». En: *Chinese Optics Letters* 10.s1 (2012), S10606.
- [29] *Tensor Flow*. <https://www.tensorflow.org>.
- [30] Christian Von Lucken, Benjamin Baran y Carlos Brizuela. «A survey on multi-objective evolutionary algorithms for many-objective problems». En: *Computational optimization and applications* 58.3 (2014), págs. 707-756.