UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA

Modelos Deep Q-Network y su aprendizaje del control dinámico de la interferencia para mejorar el rendimiento de redes móviles celulares

TESIS

que presenta para obtener el grado de DOCTOR EN CIENCIAS

Cosme Alexis Anzaldo Navarrete

DIRECTOR DE TESIS
DR. ÁNGEL GABRIEL ANDRADE REÁTIGA

MEXICALI, B. C. **OCTUBRE DE 2023**

TESIS DEFENDIDA POR

Cosme Alexis Anzaldo Navarrete

Y APROBADA POR EL SIGUIENTE COMITÉ

_____

Dr. Ángel Gabriel Andrade Reátiga

Director del Comité

_____

Dr. Oscar Humberto Montiel Ross

Miembro del Comité

_____

Dra. Kenia Picos Espinoza

Miembro del Comité

_____

Dr. Guillermo Galaviz Yáñez

Miembro del Comité

_____

Dr. Enrique René Bastidas Puga

Miembro del Comité

_____

Dr. Wendy Flores Fuentes

Coordinadora de Posgrado e Investigación

Facultad de Ingeniería

Mexicali, Baja California, a 05 de Octubre de 2023

RESUMEN de la Tesis de Cosme Alexis Anzaldo Navarrete, presentada como requisito parcial para la obtención del grado de DOCTORADO EN CIENCIAS. Mexicali, Baja California, México, 05 de Octubre, 2023.

Modelos Deep Q-Network y su aprendizaje del control dinámico de la interferencia para mejorar el rendimiento de redes móviles celulares

Resumen aprobado por:

_____
Dr. Ángel Gabriel Andrade Reátiga
Director de tesis

Con el avance de la tecnología de comunicación móvil celular, se espera que una mayor cantidad y diversidad de dispositivos celulares coexistan en una misma área de cobertura, pero, por otro lado, los nuevos servicios y aplicaciones móviles demandarán más recursos radioeléctricos. Bajo este escenario, y con el poco espectro disponible para los nuevos sistemas móviles de quinta generación o más allá (B5G), gestionar los recursos (canales y su respectiva potencia) de la red móvil para que este sea más competente se vuelve complejo debido a la interferencia originada por la necesidad de que múltiples dispositivos utilicen la misma banda espectral. Para afrontar las exigencias del proceso de gestión de recursos, las técnicas de Inteligencia Artificial (IA) han demostrado, en este y otros contextos, que es una tecnología capaz de incrementar el rendimiento de la red móvil. En este trabajo de tesis, se implementa un modelo de IA llamado Deep Q-Network (DQN) para gestionar el nivel de potencia de transmisión de las estaciones base en respuesta a las condiciones dinámicas del entorno de propagación. La asignación inteligente de este nivel de potencia permite controlar el grado de interferencia de la red B5G, eficientizando, como resultado, el uso compartido del recurso radioeléctrico. Sin embargo, dado el comportamiento dinámico de la red B5G, la estrategia de asignación de potencia no puede ser estática, como tampoco la forma en la que se entrena el modelo DQN para la gestión de la potencia. En este sentido se propuso crear diversas experiencias a partir de la asignación de diferentes niveles de potencia de transmisión durante el entrenamiento de ajuste del modelo DQN. El objetivo es que el modelo DQN adapte sus parámetros de operación con base a lo aprendido de los distintos comportamientos del entorno. Se diseñó un protocolo de evaluación basado en transferencia sim2sim, con el que se transfiere el conocimiento aprendido de los modelos DQN entre los entornos de simulación. Para reducir el tiempo en el que el modelo DQN se adapta a las condiciones de red y las variaciones en la capacidad durante el entrenamiento de ajuste, se implementó un mecanismo de doble buffer que permite preservar el conocimiento aprendido de distintos entornos de red mientras se reutiliza el conocimiento actual del entorno de red. Los resultados muestran que el tiempo transitorio se reduce durante los entrenamientos de ajustes del modelo DQN. Además, se incrementa la confiabilidad del modelo DQN y se reduce la variación de la capacidad durante la asignación de potencia.

Palabras clave: Aprendizaje por refuerzo profundo, asignación de potencia, B5G, buffer de repetición de experiencias, redes inalámbricas.

ABSTRACT of the thesis, presented by Cosme Alexis Anzaldo Navarrete, in order to obtain the DOCTOR IN SCIENCE DEGREE. Mexicali, Baja California, México, October 5th, 2023.

Deep Q-Network models and their learning of dynamic interference control to improve the performance of cellular mobile networks

Approved by:

_____
Dr. Ángel Gabriel Andrade Reátiga
Thesis Advisor

With the advances in mobile cellular communication technology, a greater quantity and diversity of cellular devices are expected to coexist in the same coverage area however, on the other hand, new mobile services and applications will demand more radio frequency resources. Under this scenario, and with limited spectrum available for new Beyond Fifth-Generation (B5G) mobile systems, managing the resources (channels and their respective power) of the mobile network to make it more efficient becomes complex due to the interference originated from the need for multiple devices to use the same spectral band. Artificial Intelligence (AI) techniques have demonstrated their ability to enhance the performance of the mobile network in this and other contexts, addressing the demands of the resource management process. In this thesis work, an AI model called Deep Q-Network (DQN) is implemented to manage the transmission power level of base stations in response to dynamic propagation environment conditions. The intelligent allocation of this power level allows controlling the degree of interference in the B5G network, thereby efficiently optimizing the shared use of radio frequency resources. However, given the dynamic behavior of the B5G network, the power allocation strategy cannot be static, nor can the way the DQN model is trained for power management. In this regard, the proposal was to create various experiences based on the allocation of different levels of transmission power during the DQN model's adjustment training. The goal is for the DQN model to adapt its operational parameters based on what it has learned from different environmental behaviors. An evaluation protocol based on sim2sim transfer was designed, through which the knowledge learned from the DQN models is transferred between simulation environments. A double buffer mechanism was implemented to reduce the time required for the DQN model to adapt to network conditions and variations in capacity during adjustment training. This mechanism preserves the learned knowledge from different network environments while reusing the current knowledge of the network environment. The results show a reduction in transient time during DQN model adjustment training. Furthermore, the reliability of the DQN model increases, and capacity variation during power allocation decreases.

Keywords: Deep reinforcement learning, power allocation, B5G, experience replay buffer, wireless networks.

AGRADECIMIENTOS

A mi novia Carolina por apoyarme siempre y estar a mi lado en todo momento.

Al Dr. Ángel G. Andrade Reátiga por su tiempo, enseñanzas y orientación brindadas durante mis estudios de doctorado.

A los miembros de mi comité de tesis, Dr. Oscar Humberto Montiel Ross, Dra. Kenia Picos Espinoza, Dr. Guillermo Galaviz Yáñez y Dr. Enrique René Bastidas Puga por sus recomendaciones y ayuda para mejorar mi trabajo de investigación.

Al Consejo Nacional de Humanidades, Ciencias y Tecnología (CONAHCYT) por apoyarme económicamente durante mis estudios de doctorado.

A la Universidad Autónoma de Baja California (UABC), por darme la oportunidad de seguir con mis estudios.

ÍNDICE

Página

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La industria de las comunicaciones móviles celulares se desarrolla constantemente, de manera que las tasas de transferencia de datos y el ancho de banda que se ofrecen a los usuarios se incrementan en los sistemas de comunicación móvil conforme avanzan las generaciones. Este avance promueve mejoras a distintos sectores, como el de transporte, la salud, la agricultura y las finanzas. A medida que estas redes móviles evolucionan es de esperarse que una mayor cantidad y diversidad de dispositivos inalámbricos coexistan en una misma área de cobertura, los servicios móviles demanden más recursos radioeléctricos y se adopten nuevos paradigmas de comunicación. Este progreso trae consigo retos que necesitan atenderse, tales como, reducir el consumo de energía de las estaciones base (BS – base station) y de los equipos de usuarios celulares (UE – user equipment) y controlar la interferencia entre los BS y UE, quienes comparten porciones de espectro, entre ellos y con otras tecnologías, como una estrategia para mejorar la eficiencia espectral. Estos desafíos se vuelven complejos de resolver frente al incremento de UE y de BS móviles, cuyas altas demandas de tasas de transferencia de datos y bajas exigencias de retardo se esperan para las futuras aplicaciones móviles [1]. Para estos sistemas de comunicación móvil de nueva generación, como los de quinta y sexta generación (denominados B5G) [1], [2], [3], la Inteligencia Artificial (IA) se considera como una tecnología fundamental para que puedan enfrentar los distintos desafíos que las nuevas aplicaciones impondrán, tales como, la conectividad vehicular, la comunicación máquina a máquina, la realidad virtual y aumentada,

las comunicaciones holográficas o el internet táctil [4]. Sin embargo, es un reto crear modelos de IA para implementarlos en entornos de comunicación móvil del mundo real, debido a la cantidad y dinámica de los escenarios previstos en las redes B5G. Para automatizar y eficientizar los procesos de la red celular mediante técnicas de IA, es necesario diseñar mecanismos que logren adaptarse, en el menor tiempo posible, a las dinámicas de la red móvil.

1.1 Contexto del problema

Aumentar la capacidad de una red de comunicaciones móvil depende principalmente de dos factores: (i) incrementar el ancho de banda a cada servicio móvil y (ii) reducir la interferencia que perciben los nodos en la red inalámbrica. Dado que el espectro radioeléctrico para las comunicaciones móviles es limitado, su reutilización y compartición, mediante el despliegue de una gran cantidad de nodos de baja potencia (estaciones base pequeñas (SBS - small base station), es una de las soluciones más abordadas en la literatura [5]. Sin embargo, reutilizar y compartir el espectro radioeléctrico genera señales que interfieren no solo a los nodos localizados en la misma celda, sino también a otros en celdas cercanas. Esto, en consecuencia, afecta la calidad de los enlaces de comunicación. Por ejemplo, en la Figura 1.1 se muestra la interferencia que percibe un UE desde la estación base central (MBS – macro base station) y por dos SBS cercanas. Con el incremento de las SBS y la reutilización del espectro se vuelve complicado mantener la calidad de servicio (QoS – quality of service) de los enlaces de comunicación (ascendente y descendente) debido al incremento de interferencia. Por lo que es fundamental controlar la interferencia en los sistemas B5G ultradensificados.

1.1.1 Técnicas de optimización para la asignación de recursos

La asignación de recursos es un tema que se ha estudiado desde los sistemas móviles de segunda generación (2G) y ha evolucionado conforme la QoS, el retardo y la tasa mínima de transmisión de datos se han modificado a partir de las necesidades de las aplicaciones móviles. En sistemas inalámbricos en los que no se consideran la compartición o reutilización del espectro, el nodo central de la red se encarga de asignar los enlaces de comunicación a cada nodo para evitar que dos nodos transmitan en un mismo canal. Lo anterior permite

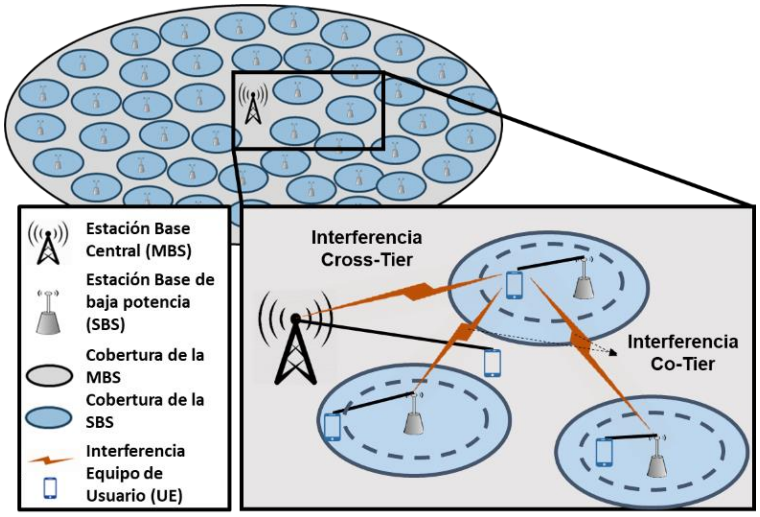

Figura 1.1. Representación de los tipos de interferencia en una red celular móvil.

establecer enlaces libres de interferencia y lograr una transmisión fiable entre el origen y el destino. En este caso, no es necesario controlar la potencia que utiliza cada UE para transmitir su información. Para decidir cómo deben distribuirse los recursos entre los UE, los algoritmos de asignación de recursos toman en cuenta una o varias métricas de desempeño como incrementar la capacidad, calidad, o equidad de la red o reducir el retardo durante la transmisión de información.

Para las redes B5G, en las que la compartición y re-uso de espectro es una necesidad, es necesario implementar técnicas eficientes de asignación de recursos (canal y potencia) para mitigar la interferencia, y con ello, garantizar la calidad de servicio y las tasas mínimas de transmisión de datos requeridas por los UE. Además, el tiempo de cómputo para realizar la asignación de recursos debe calcularse en menos de 1 milisegundo, según lo establecido por el estándar B5G [1]. Debido a esto, la complejidad de los algoritmos de asignación de recursos no debe ser elevada; de lo contrario, afectará la capacidad de la red. Paradójicamente, la asignación de estos recursos se formula como un problema de programación no lineal de enteros mixtos, que es no-convexo y ‘‘Np-hard’’ [6]. Por lo que encontrar soluciones óptimas es un proceso computacionalmente complejo. Los algoritmos de optimización global aumentan su complejidad debido a las condiciones dinámicas de la red, como la movilidad de los UE, los cambios en la información de estado del canal (CSI – channel state information), el número de BS o UE y sus requisitos de QoS. Por otra parte, los algoritmos de menor complejidad, como los algoritmos heurísticos, tienen la desventaja de

logar un rendimiento menor en comparación con los algoritmos de última generación (i.e., algoritmos del estado del arte), como la Programación Fraccionada (FP – fractional programming) [7] y el Error Cuadrático Medio Mínimo Ponderado (WMMSE – weighted mean mínimum square error) [8]. Encontrar la solución al problema de asignación de recursos por medio de estos algoritmos de optimización requiere de un proceso de búsqueda iterativo y evaluar cada condición del entorno de red. Este proceso de búsqueda limita la implementación de estos algoritmos de optimización en sistemas reales debido a que su tiempo de ejecución es mayor que el tiempo en que las condiciones del canal se mantienen estables. Es decir, la calidad de la solución se degrada debido a que la solución encontrada por el algoritmo de optimización ya no coincide con la condición del entorno al momento de implementarse. Además, estos algoritmos requieren información de la red, como el Indicador de Intensidad de la Señal Recibida (RSSI – received signal strength indicator), la Relación Señal-Interferencia-más-Ruido (SINR – signal-to-interference-plus-noise-ratio) y los recursos de radiofrecuencia disponibles en toda la red. A pesar de que esta información puede obtenerse en cualquier momento, conseguirla de forma instantánea y precisa se vuelve inviable debido a que se requiere recibir la información desde todas las entidades que se encuentran distribuidas en la red. Lo anterior, limita la eficiencia de los métodos de optimización para encontrar una adecuada asignación de recursos [9]. Los enfoques de Aprendizaje Automático (ML – machine learning) pueden obtener soluciones en poco tiempo [10] e incluso adaptarse a las condiciones que no se tomaron en cuenta en los modelos simulados. En otras palabras, los algoritmos de ML encuentran soluciones de forma más rápidas con información parcial de la red [11], [12].

1.1.2 Aprendizaje automático para la asignación de recursos

Con base a una metodología de revisión sistemática de la literatura (Ver Capítulo 2), identificamos que la mayoría de los trabajos de investigación que implementan técnicas de ML para la asignación de recursos en redes celulares de alta densificación de SBS [11], [13-18] utilizan conjuntos de datos sintéticos, creados a partir de modelos de simulación. Aunque estos modelos muestran cierta solidez frente a condiciones de red no consideradas en sus conjuntos de datos sintéticos, no es factible imitar la dinámica del mundo real en modelos de simulación. Una de las características de las redes móviles B5G es que operarán en un

entorno dinámico (por ejemplo, movilidad de UE y movilidad de BS, variaciones de las demandas de QoS, coexistencia de diferentes tecnologías inalámbricas) por lo que es necesario que las técnicas de ML se adapten de manera autónoma y continua a las condiciones de red conforme se presenten.

Los modelos de ML apropiados para adaptarse satisfactoriamente a las fluctuaciones de las redes móviles son los denominados Aprendizaje por Refuerzo Profundo o Deep Reinforcement Learning (DRL). Los modelos DRL generan sus propios conjuntos de datos conforme interactúan con el entorno. Esta característica les permite adaptarse a los comportamientos inesperados causados por las dinámicas del entorno de la red celular, pero requieren de un entrenamiento eficiente para que aprendan la mejor estrategia de asignación de recursos [19].

1.1.3 Entrenamiento de los modelos de aprendizaje por refuerzo profundo

El proceso de entrenamiento de los modelos DRL consiste en explorar y explotar las dinámicas del entorno mediante la recopilación y el uso de las muestras de datos [20]. En este trabajo, llamamos experiencias a los datos que generan los modelos DRL y que representan distintos escenarios y condiciones que pueden presentarse en las redes B5G. Estas experiencias describen la interacción entre un agente (es decir, la entidad que ejecuta el modelo DRL) y el entorno del sistema. Con base a estas experiencias, se actualizan los parámetros del modelo DRL para encontrar la política óptima de asignación de recursos. La política es la estrategia que dicta las acciones que el agente ejecuta en función de la información actual del entorno para alcanzar el objetivo (por ejemplo, maximizar la capacidad de la red o la cantidad de usuarios atendidos).

El ciclo de interacción entre el agente y el entorno que genera una experiencia se muestra en la Figura 1.2. El agente recibe el estado (es decir, la información actual del entorno) y con base a esta información ejecuta una acción siguiendo la política del modelo DRL. Una acción puede ser la asignación de un canal a un UE o la asignación del nivel de potencia de transmisión a una BS. A partir de dicha la acción, el agente recibe una retroalimentación sobre el efecto provocado en el rendimiento de la red. La acción, el estado y la recompensa involucrados en la interacción entre el agente y el entorno forman una

experiencia. Estas experiencias se almacenan en un buffer de repetición (ER – experience replay) que constituyen el conjunto de datos en los modelos DRL [21].

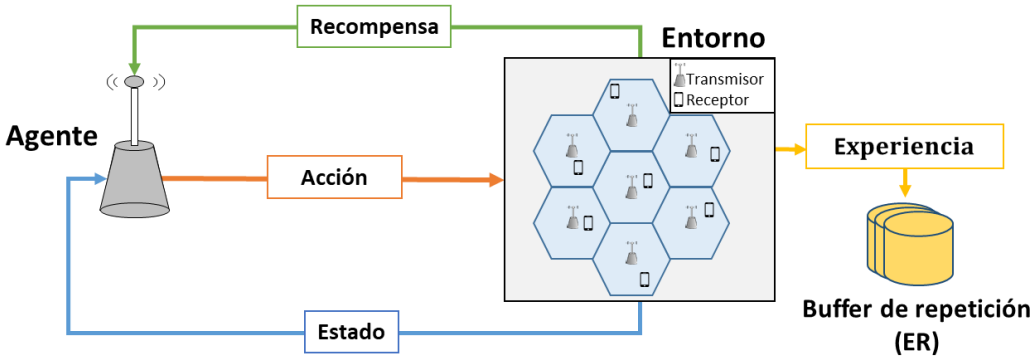

Figura 1.2. Ciclo de interacción de una estación base de baja potencia para generar una experiencia en los modelos DRL.

1.1.4 Brecha de realidad de los modelos de aprendizaje por refuerzo profundo

A pesar de la adaptabilidad que logran las entidades de la red por implementar agentes entrenados para configurar los parámetros de los sistemas B5G (por ejemplo, inclinación de antena o potencia de transmisión), se requieren analizar los efectos que producen los modelos DRL en el rendimiento de la red ante escenarios dinámicos. Implementar modelos DRL bajo escenarios que cambian de un momento a otro, puede causar incertidumbre en el desempeño esperado del sistema al tratar de adaptarse durante la fase de entrenamiento. Por ejemplo, algunos agentes (SBS) de la red podrían enfocarse en priorizar, ya sea, la latencia, capacidad o calidad de experiencia, mientras que otros ni siquiera estén conectados a la red. Por tal motivo, se vuelve relevante considerar casos como sim2sim [22] o sim2real [23] para analizar los efectos que conlleva la transferencia de conocimiento de los modelos entrenados. En sim2sim se transfiere el conocimiento de un entorno de simulación a otro entorno de simulación, mientras que sim2real transfiere el conocimiento de un entorno de simulación a un entorno real.

El proceso de implementación de las técnicas de DRL se presenta en la Figura 1.3. Primero se realiza un entrenamiento bajo simulación considerando diferentes condiciones de red. Después, se realiza un entrenamiento de ajuste, en el que se utilizan conjuntos de datos que representen escenarios del entorno real en donde el modelo será implementado. En caso de contar con datos previos del entorno real es posible implementar la estrategia mostrada en la Figura 1.3a, en la que el modelo puede entrenarse de nuevo antes de implementarlo en los

entornos reales. En caso contrario, se implementa la estrategia 1.3b, en la que el agente genera su propio conjunto de datos al interactuar con el entorno de implementación. Durante esta interacción el agente ejecuta acciones al azar para generar experiencias que le permitan ajustar el modelo al nuevo entorno de implementación. Estas interacciones en las que el modelo DQN recopila experiencias y actualiza sus parámetros operación para adaptarse a los cambios del entorno se denomina en esta tesis como entrenamiento de ajuste. A medida que se modifican las condiciones del entorno de la red B5G, los modelos DRL requerirán entrenamientos de ajuste eficientes para evitar que se degrade el rendimiento de la red durante la exploración de nuevos entornos.

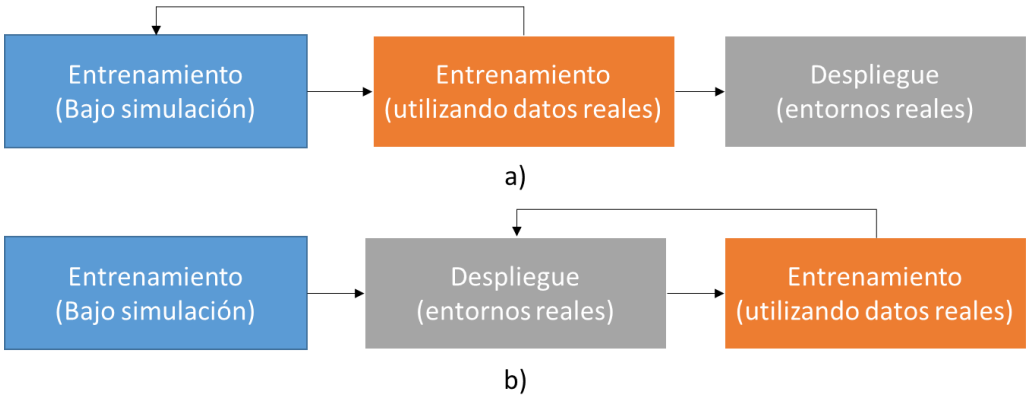

Figura 1.3. Proceso de transferencia de conocimiento de los modelos DRL entrenados bajo simulación a entornos reales. a) Proceso con datos disponibles del entorno de implementación. b) Proceso sin datos disponibles del entorno de implementación.

1.2 Planteamiento del problema

Para lograr modelos DRL que provean soluciones precisas en el momento indicado, es necesario entrenarlos con cientos de datos que representen diversas condiciones del entorno. Incluso, con tal diversidad de experiencias contenidas en el buffer de ER, es inviable contar con un modelo de simulación que genere todos los escenarios posibles que pudieran presentarse en el mundo real. Aquellos escenarios imprevistos provocan que las soluciones que ejecute el modelo previamente entrenado no sean apropiadas para lograr las capacidades esperadas bajo las nuevas condiciones. Por lo tanto, un agente entrenado en un tiempo anterior requerirá, en un tiempo posterior, de un entrenamiento de ajuste para adaptarse a la política del modelo DRL con las nuevas condiciones del entorno de red. En la Figura 1.4 se puede observar el efecto negativo que se presenta en el rendimiento de la red al momento en

que se presenta un cambio inesperado en el entorno de la red (fluctuación de la red, movilidad de usuarios, demandas de tráfico, etc.). En este caso, el modelo dedica varios instantes de tiempo ejecutando acciones aleatorias para aprender una política estable (sub-óptima). A este tiempo de aprendizaje se le denomina tiempo transitorio [19]. Se observa en la figura que durante el tiempo transitorio se presentan variaciones en el rendimiento de la red provocados por la ejecución de acciones aleatorias y las fluctuaciones de las condiciones de la red. Por lo anterior, el entrenamiento de ajuste se vuelve costoso y compromete la seguridad y confiabilidad del sistema inalámbrico. Primero, porque se requiere una transmisión adicional de mensajes de señalización para propagar el CSI dedicados a actualizar el modelo a través de la cooperación entre las BS. En segundo lugar, las acciones aleatorias, debido a la exploración, aumentan la probabilidad de generar interferencia en el sistema, degradando el rendimiento de la red. Por último, por el desperdicio de recursos computacionales que se produce al ejecutar un entrenamiento adicional. Por lo tanto, es necesario reducir el tiempo transitorio y la variación del rendimiento de la red durante el entrenamiento de ajuste.

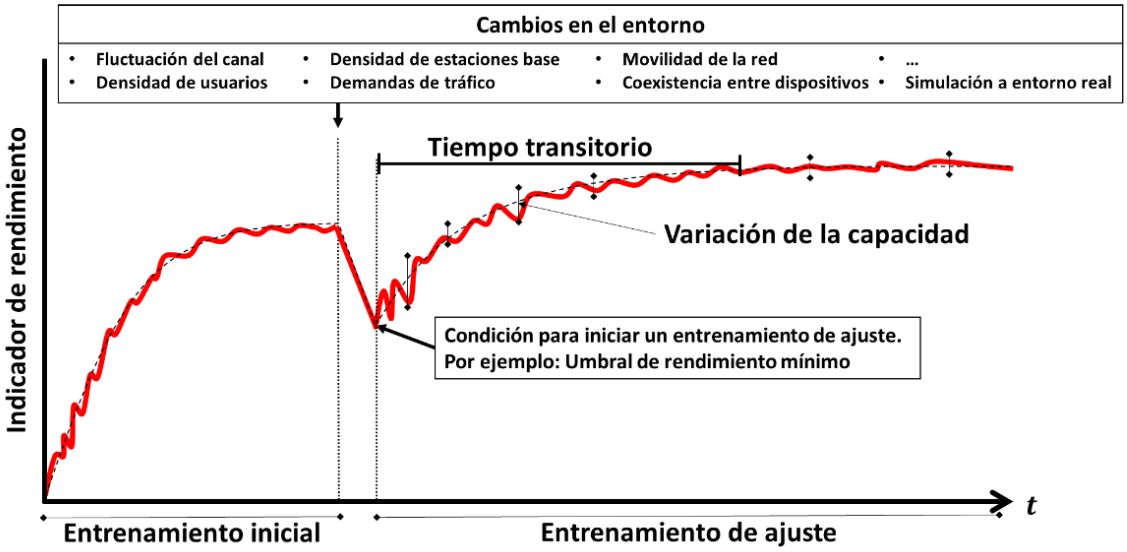

Figura 1.4. Curva de entrenamiento de ajuste de un modelo de aprendizaje por refuerzo profundo debido a un cambio en el entorno.

1.2.1 Gestión de experiencias del buffer de repetición

Uno de los desafíos del entrenamiento de ajuste es la estrategia de exploración. La estrategia de exploración se refiere a la decisión que toma el agente respecto a su acción a ejecutar durante el entrenamiento del modelo. Es decir, durante la fase del entrenamiento de

ajuste el agente DRL tiene dos opciones: 1) seguir la política actual, seleccionando acciones adecuadas para las condiciones del entorno previas o, 2) explorar el entorno, seleccionando nuevas acciones de forma aleatoria. Seguir la política actual introduce un sesgo en el buffer de ER, lo que provoca que el buffer de ER carezca de diversidad en el espacio de acción, ya que se duplican las acciones siguiendo la política previa. Por otro lado, ejecutar las acciones aleatorias provoca pérdidas en el rendimiento del sistema, pero añade diversidad en el espacio de acción, lo que puede llevar a que la política se adapte mejor a las condiciones del entorno actual.

Una de las formas de evitar la degradación el rendimiento de la red durante el entrenamiento de ajuste es mediante la gestión del buffer de ER [21]. Es decir, reutilizar las experiencias ya generadas en lugar de ejecutar acciones aleatorias para generar nuevas experiencias. Esto incrementa la eficiencia de las experiencias del buffer de ER y evita realizar acciones catastróficas. Sin embargo, el buffer de ER requiere mecanismos eficientes para gestionar estás experiencias. La gestión del buffer de ER consiste en mecanismos de retención y muestreo, para seleccionar las experiencias que se van a almacenar y las experiencias que se utilizarán durante el aprendizaje, como se muestra en la Figura 1.5.

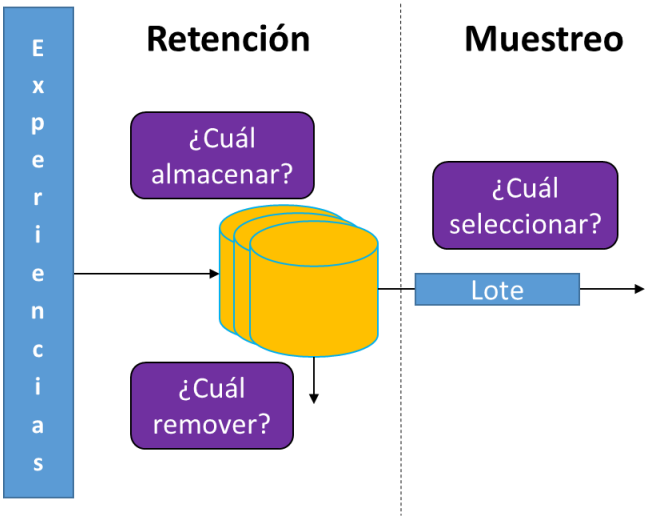

Figura 1.5. Mecanismos de gestión del buffer de repetición.

1.2.2 Retos de la gestión de experiencias del buffer de repetición en redes celulares

Durante el entrenamiento de ajuste, el buffer de ER no contiene experiencias del nuevo entorno por lo que se vuelve inevitable que el agente ejecute acciones al azar para explorar el entorno. En la literatura, generalmente se gestiona el buffer de ER mediante el

esquema de gestión FIFO, con el que se almacenan las experiencias más nuevas reemplazando a las más antiguas [21]. Aunque esta gestión de experiencias es efectiva para un entrenamiento inicial [11], [24], para explorar el nuevo entorno durante el entrenamiento de ajuste, las acciones aleatorias se ejecutan de forma limitada para evitar caídas del rendimiento de la red. Es decir, las experiencias generadas por la exploración de los modelos DRL son limitadas. Esto provoca que la diversidad de las experiencias del buffer de ER se pierda ya que la gestión de experiencias basada en FIFO retiene estas nuevas experiencias por poco tiempo. Una forma de mantener la diversidad en el buffer de experiencias es mediante la gestión de experiencias del buffer de ER. Sin embargo, identificar las experiencias relevantes para las redes inalámbricas es un desafío debido al comportamiento estocástico del canal inalámbrico. Por ejemplo, la tasa de datos del sistema cambia constantemente ya que depende de las condiciones del entorno espacio-temporal. Por consiguiente, aplicar mecanismos para retener experiencias con valores de recompensa altos preservará el conocimiento de condiciones de red que propicien el mayor rendimiento de la red. Por lo que el modelo DRL no retendrá conocimiento de condiciones más desafiantes en donde el valor de recompensa será más bajo (por ejemplo, desvanecimientos profundos o UE en los bordes de las celdas), lo que sesgará el buffer de ER y fallará en la generalización de la política [25]. Por otra parte, los mecanismos del buffer de ER que asignan un valor de prioridad a cada experiencia con base a su relevancia durante el aprendizaje requieren actualizar constantemente los valores de prioridad de cada experiencia [26]. Considerando que múltiples agentes en el sistema celular estarán generando experiencias, el proceso para actualizar los valores de prioridad se vuelve computacionalmente costoso. Asimismo, mecanismos que incorporan diversidad en las experiencias requieren un cálculo adicional para determinar el valor priorizado de la experiencia a partir del cual seleccionarlas [27], [28] o removerlas [29-31] del buffer de ER.

Por lo tanto, en este trabajo se propone un mecanismo de retención de experiencias de doble buffer. En uno de los buffers se retienen las experiencias de exploración, mientras que en el otro se retienen las de explotación. Esta estrategia retiene las experiencias de exploración agregando diversidad a las experiencias seleccionadas durante el entrenamiento de ajuste, mientras que el buffer de explotación permite renovar las experiencias al retener las experiencias enfocadas a la política del entorno actual.

1.3 Pregunta de investigación

Implementar la mejor política de asignación de recursos para un escenario particular (por ejemplo, redes zonas comerciales, zonas académicas, comunicaciones dispositivo a dispositivo, entre otros) por medio de los modelos DRL se vuelve un reto debido a la dinámica de la red móvil. Por ejemplo, la coexistencia de diferentes tecnologías y diversos dispositivos inalámbricos generando múltiples condiciones de operación para la red, tales como, fluctuaciones del canal de propagación, despliegue de nuevos nodos, conexión y desconexión de usuarios en distintas áreas de cobertura o distintos requisitos de calidad de servicio. Para adaptarse a las nuevas condiciones de la red, los modelos DRL deben explorar estos nuevos entornos tomando diferentes políticas de asignación de recursos (i.e., fase de exploración de los modelos DRL) para aprender las nuevas dinámicas del entorno y actualizar los parámetros del modelo hacia la política de asignación de recursos óptima. En lugar de ejecutar acciones aleatorias para explorar el nuevo entorno, una estrategia es sesgar la política utilizando el conocimiento de las experiencias actuales o de otros modelos bajo circunstancias de entorno similares durante el proceso de entrenamiento con el fin ejecutar acciones que fueron adecuadas para entornos semejantes. Sin embargo, en la literatura actual no se aborda cómo identificar y administrar el conocimiento de los modelos DRL en problemas de asignación de recursos para redes inalámbricas. Por lo tanto, en este trabajo se considera el caso de transferencia sim2sim como primer paso para la transferencia sim2real en redes celulares B5G con el fin analizar los efectos del tiempo transitorio y la variación del rendimiento de la red ocasionados durante el entrenamiento de ajuste. Para mitigar los efectos negativos durante el entrenamiento de ajuste, se propone implementar un mecanismo de retención de experiencias de doble buffer para retener y reutilizar por mayor tiempo aquellas experiencias que se generaron durante la exploración de nuevas condiciones de entornos de red. Por consiguiente, esta tesis tiene como objetivo responder la siguiente pregunta de investigación:

- ¿De qué manera influye el mecanismo de gestión de experiencias basado en doble buffer en el tiempo transitorio y en la estabilidad de la capacidad de la red durante el aprendizaje de nuevas políticas de asignación de potencia en redes móviles celulares?

1.4 Hipótesis

- Hipótesis nula 1 ($H1_0$): La combinación de un mecanismo de gestión de experiencias y la reutilización de experiencias adquiridas bajo diversas condiciones (ubicación de UE y CSI) no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia en redes móviles B5G.
- Hipótesis alternativa 1 ($H1_1$): La combinación de un mecanismo de gestión de experiencias y la reutilización de experiencias adquiridas bajo diversas condiciones (ubicación de UE y CSI) reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia en redes móviles B5G.
- Hipótesis nula 2 ($H2_0$): La implementación de un mecanismo de gestión de experiencias que retenga las experiencias en dos buffers independientes (de exploración y de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia.
- Hipótesis alternativa 2 ($H2_1$): La implementación de un mecanismo de gestión de experiencias que retenga las experiencias en dos buffers independientes (de exploración y de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia.
- Hipótesis nula 3 ($H3_0$): La implementación de un mecanismo de gestión de experiencias que retenga experiencias en dos buffers independientes (i.e., buffer de exploración y buffer de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá la variación de la capacidad de la red durante la asignación de potencia en los modelos Deep Q-Network.
- Hipótesis alternativa 3 ($H3_1$): La implementación de un mecanismo de gestión de experiencias que retenga experiencias en dos buffers independientes (i.e., buffer de exploración y buffer de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G reducirá la variación de la capacidad de la red durante la asignación de potencia en los modelos Deep Q-Network.

1.5 Objetivo General

Diseñar un mecanismo de gestión de experiencias para modelos Deep Q-Network que reduzca el tiempo transitorio y la variabilidad de la capacidad durante el aprendizaje de nuevas condiciones de red con aplicación en la asignación de potencia en redes móviles B5G.

1.6 Objetivos específicos

- Desarrollar un protocolo de evaluación que permita evaluar el tiempo transitorio y la variabilidad de la capacidad que logran los modelos Deep Q-Network durante el aprendizaje de nuevas condiciones de asignación de potencia en redes móviles B5G.
- Controlar la potencia en una red móvil B5G a partir de un mecanismo de reutilización de experiencias y de experiencias aprendidas por los modelos Deep Q-Network.
- Reducir el tiempo transitorio y la variabilidad de la capacidad en un modelo de aprendizaje Deep Q-Network a partir de la creación de un sistema de buffers independientes (exploración y explotación) como parte del mecanismo de gestión de experiencias adquiridas y aprendidas durante el aprendizaje de nuevas condiciones de asignación de potencia en redes móviles B5G.
- Evaluar los efectos en la capacidad de la red móvil B5G y en el tiempo transitorio por reutilizar experiencias previamente aprendidas por los modelos Deep Q-Network en distintas condiciones de la red.
- Evaluar los efectos en la capacidad de la red móvil B5G y en el tiempo transitorio a partir de la implementación de un mecanismo de gestión de experiencias de doble buffer durante el aprendizaje de distintas condiciones de la red utilizando el modelo Deep Q-Network para el control de potencia.
- Analizar los efectos que se presentan en el tiempo transitorio y en la variación de la capacidad durante el aprendizaje de nuevas condiciones de asignación de potencia en redes móviles B5G cuando se utilizan dos buffers de experiencias independientes y se reutilizan las experiencias de modelos Deep Q-Network previamente entrenados.

1.7 Aportaciones de la tesis

- Desarrollo e implementación de un modelo DQN para la asignación inteligente de potencia en estaciones base en redes B5G. Este modelo es diseñado para adaptarse a las cambiantes condiciones del entorno y lograr un mejor rendimiento en la asignación de recursos radioeléctricos.
- Desarrollo de un protocolo de evaluación que permite medir el tiempo transitorio y la variabilidad de la capacidad en los modelos DQN durante el aprendizaje de nuevas condiciones de red para la asignación de potencia en redes móviles B5G.
- Método para acelerar el entrenamiento mediante un mecanismo de gestión de experiencias basado en un sistema de dos buffers. Esta estrategia permite preservar la diversidad en el buffer de experiencias adquiridas durante el entrenamiento de ajuste resultando en una reducción del tiempo transitorio y una reducción en la variabilidad de la capacidad de la red durante el aprendizaje de nuevas condiciones de red para la asignación de potencia en redes móviles B5G.

1.8 Estructura de la tesis

Esta tesis se encuentra organizada de la siguiente manera:

En el capítulo 2 se presenta una revisión sistemática de la literatura de los trabajos que implementan Aprendizaje Automático (ML) para la asignación de recursos en Redes Ultra-Densas (UDN). Además, se discuten los problemas abiertos identificados de los trabajos relacionados.

El capítulo 3 describe el esquema de Aprendizaje por Refuerzo Profundo (DRL) implementado. Los elementos que conforman el modelo Deep Q-Network (DQN) descritos son la red neuronal profunda, la red objetivo, el buffer de repetición y la estrategia de exploración del modelo. Por último, se examina la transferencia del conocimiento para aprovechar los conocimientos previos de otros modelos DQN.

En el capítulo 4 se presentan los mecanismos de gestión de experiencias implementados para los modelos DQN y se describe el esquema de aprendizaje utilizado para resolver el problema de asignación de potencia. Además, se presentan las métricas de desempeño para evaluar el tiempo transitorio y la variación de la capacidad durante el entrenamiento de ajuste.

En el capítulo 5 de la tesis se presentan los resultados a través de una serie de experimentos. En los experimentos se analizan los efectos en el aprendizaje del tamaño del buffer de Repetición de Experiencias (ER). Además, se analizan y discuten los efectos de la diversidad en el buffer de ER al utilizar el mecanismo de Repetición de Experiencias Dual (DER) propuesto en comparación con los mecanismos de gestión de la literatura.

Por último, en el capítulo 6 se presenta el resumen y la discusión de los resultados obtenidos en los experimentos realizados. Además, en este capítulo se responde la pregunta de investigación y se sugieren direcciones para futuros trabajos.

Capítulo 2

Revisión sistemática de la literatura

Las redes B5G tienen la meta de aumentar la tasa de datos, disminuir el retardo e incrementar la atención a usuarios en comparación con los sistemas móviles de generaciones previas. Sin embargo, el incremento constante de suscriptores móviles junto con el espectro radioeléctrico a punto de saturarse limitará que se cumpla. Para incrementar la eficiencia espectral en los sistemas B5G, se ha propuesto el despliegue masivo de nodos de baja potencia sobre la cobertura de la red celular existente. Los nodos pueden ser estaciones base pequeñas (SBS- small base station), drones o satélites de retransmisión. Esta densificación de radio bases acortará la distancia de los enlaces de comunicación y, con esto, beneficiará a un mayor re-uso del espectro, mejorará la calidad de los enlaces y permitirá que puedan usarse señales a frecuencias muy altas (en el orden de las ondas milimétricas). Sin embargo, esta compartición de bandas espectrales entre nodos de distintas tecnologías (satélites, móviles, o bandas no-reguladas) requerirá de una asignación eficientemente de los recursos (potencia y ancho de banda) para controlar la potencial interferencia entre las transmisiones coexistentes. Con el incremento de usuarios móviles y radio bases, la asignación de recursos se vuelve un problema complejo de resolver con algoritmos de aproximación, debido a la dimensionalidad del sistema, al amplio espacio de búsqueda y a la reutilización agresiva del espectro. Como se mencionó en el capítulo anterior, el uso de técnicas de IA será una de las características de los sistemas B5G. Una dirección prometedora para atender los desafíos

descritos anteriormente es adoptar tecnologías de aprendizaje automático para analizar y administrar los recursos en los sistemas B5G.

Con base a una revisión preliminar de la literatura, se identificó que las técnicas de IA (principalmente de aprendizaje automático) se han aplicado en distintos tipos de redes inalámbricas, como redes Dispositivo a Dispositivo (D2D), redes de internet de las cosas (IoT), redes de sensores inalámbricos (WSN) y redes celulares para resolver problemas de descarga de tráfico, asociación de UE (UEA – user equipment association) o la selección de BS. Sin embargo, se encontraron pocos documentos que abordan el uso de técnicas de aprendizaje automático para resolver el problema de asignación de recursos en redes con características de redes ultra-densas (UDN). Esta ausencia de conocimiento motivó la revisión sistemática de literatura que se presenta en este capítulo.

2.1 Diseño del estudio

En esta sección se presenta la metodología que se siguió para realizar esta revisión sistemática. La estructura se basa principalmente en [32], que describe las guías para realizar revisiones sistemáticas en ingeniería de software. La metodología consta de tres etapas: planificación, realización y presentación de informes. En la Sección 2.1 se describe la etapa de planificación en el que se establece la necesidad de contar con una revisión sistemática y se describe el diseño del protocolo. Después, la etapa de realización consiste en seleccionar los trabajos, con base a criterios de selección y extracción de datos, descritos en la Sección 2.2. Finalmente, los resultados de la extracción de datos de los documentos seleccionados se discuten en la Sección 2.3, y la conclusión de los hallazgos identificados se presenta en la Sección 2.5.

2.1.1 Trabajos relacionados

En la Tabla 2.1 se enlistan los siete documentos que se identificaron como del tipo revisión sistemática. Cada uno de los trabajos se etiquetó de acuerdo al tema que abordan; aprendizaje automático, redes ultra-densas y asignación de recursos. Como se muestra en la Tabla 2.1, ninguno de los trabajos aborda específicamente la aplicación de técnicas de ML para resolver el problema de asignación de recursos en redes UDN. Por ejemplo, en [33], los autores evalúan el rendimiento de una red UDN desde la perspectiva de la interferencia, la

movilidad y el costo computacional. Los autores presentan y discuten los efectos que provoca la interferencia en la UDN, diferentes dominios de radio y la gestión de recursos y movilidad. En este trabajo no se presentan características u operación de las técnicas de ML, pero sí se destacan ejemplos en los que la gestión de recursos mejora algún indicador de rendimiento de la UDN, como el consumo de energía de la BS, las demandas de tráfico y las condiciones geoespaciales (por ejemplo, edificios) que afectan la planificación del despliegue de la BS. Los autores en [34] centran su investigación en la implementación de ML en redes futuras. En esta revisión resaltan las aplicaciones de ML en 5G (e.g., para la agrupación de celdas) y la necesidad de desarrollar técnicas de RA inteligentes capaces de tomar decisiones ante las condiciones dinámicas de la red (e.g., movilidad de usuarios). Sin embargo, no se realiza ningún análisis adicional sobre RA en UDN.

Tabla 2.1. Trabajo relacionado. ML: Aprendizaje automático. UDN: Red ultra-densa. RA: Asignación de recursos. ✗: No atendido. P: Atendido parcialmente. ✓: Atendido.

Trabajos de investigación tipo revisión sistemática	ML	UDN	RA
Ultra-Dense Networks: Survey of State of the Art and Future Directions [33]	✗	✓	P
Machine Learning Paradigms for Next-Generation Wireless Networks [34]	✓	✗	P
Resource Allocation for Ultra-Dense Networks: A Survey, Some Research Issues and Challenges [35]	P	✓	✓
Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges [36]	✓	✗	✓
Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues [37]	✓	P	✓
Applications of Deep Reinforcement Learning in Communications and Networking: A Survey [38]	✓	P	✓

Por otro lado, en [35] se describen algunos casos en los que se han aplicado algoritmos RA en UDN. Además de la densificación de BS, consideran otras funciones emergentes como son la formación de haces, la virtualización de redes, la cooperación de redes, el almacenamiento en caché o la recolección de energía. Se describen diferentes ejemplos de UDN de acuerdo con las capacidades del transmisor o receptor, como Internet de las cosas (IoT) masivo, Redes de Acceso de Radio en la Nube (C-RAN – cloud radio access network), HetNets, Múltiples-Entradas-Múltiples-Salidas (MIMO – multiple input multiple output) masivo y mmWave. A pesar de que los autores discutieron ampliamente el problema de

asignación de recursos en diferentes tipos de UDN, solo se brinda una breve introducción a los métodos de IA.

En [36], los autores centran su estudio en la aplicación de técnicas de ML y Aprendizaje Profundo (DL – deep learning) para la gestión de recursos en redes IoT no-densas. Ellos discuten las ventajas y limitaciones de diferentes técnicas de gestión de recursos aplicados en redes IoT no-densas. Los usuarios de telefonía móvil celular, las MBS y SBS tienen más capacidades, transmiten con mayor potencia y exigen velocidades de datos más altas. Por lo tanto, se requieren implementar algoritmos más complejos para controlar la interferencia con el fin de cumplir con sus requisitos cuando aumenta la densidad.

En [37], se describen técnicas de gestión de recursos para el control de potencia, gestión de espectro, gestión de backhaul, gestión de caché, diseño de formación de haces y gestión de recursos de cómputo. Los autores también aportan un análisis en profundidad de las ventajas de utilizar técnicas de ML con respecto a las técnicas convencionales (métodos de aproximación o heurísticas). Sin embargo, solo algunos trabajos evaluados en UDN son mencionados. En el trabajo en [38], describen la técnica Deep Reinforcement Learning (DLR) y sus variantes, así como su aplicación para resolver problemas, de acceso a la red, almacenamiento en caché, descarga de datos y computación, seguridad de la red, conectividad, recopilación de datos, intercambio de recursos y programación, e ingeniería y enrutamiento de tráfico en redes de comunicación inalámbricas (redes IoT, RAN en la nube, redes celulares, redes de sensores inalámbricos (WSN – wireless sensor networks) o vehículos aéreos no tripulados (UAV –unmaned aerial vehicles)). A pesar del análisis exhaustivo de los estudios y la descripción bien estructurada de los modelos DRL, no se describen escenarios, limitaciones, o problemas abiertos del desarrollo de técnicas de RA para redes UDN.

2.1.2 Objetivos de la revisión sistemática

Esta revisión sistemática tiene como objetivo identificar, de los diferentes trabajos, las características de diseño que se utilizaron durante la implementación de las técnicas de ML para resolver el problema de RA en UDN. En la Tabla 2.2 se muestran las preguntas de exploración formuladas para cumplir con dicho objetivo, así como sus respectivas motivaciones.

Tabla 2.2. Preguntas de exploración y motivación de la revisión sistemática.

Pregunta de exploración	Motivación
1. ¿Cuáles estrategias utilizan los algoritmos de ML para discriminar información relevante durante el proceso de RA en UDN?	Analizar las diferentes estrategias y consideraciones de los estudios que implementan ML para resolver el problema de RA podría ayudar a encontrar las brechas y oportunidades de estos algoritmos para llevar a más investigaciones
2. ¿Cuál indicador de rendimiento (KPI) considera el algoritmo RA basado en ML para decidir cómo asignar los recursos de la red?	Los algoritmos modifican sus estrategias según los objetivos y recursos asignados. La selección de KPI podría estar relacionada con el recurso, el objetivo y la técnica de ML implementada

2.1.3 Estrategia de búsqueda

Para la fase de extracción de datos, se realizó una búsqueda de los trabajos publicados desde enero de 2010 hasta diciembre de 2022 en las siguientes bases de datos académicas:

- IEEE Explore (http://ieeexplore.ieee.org/)
- Springer Link (http://www.springer.com/)
- ACM Digital Library (http://dl.acm.org/)
- Scopus (https://www.scopus.com/)

La cadena de búsqueda que se utilizó en las bases de datos académicas consistió de los términos '*learning method*, '*resource*', '*management*' y '*ultra-dense network*', los cuales indican la implementación de una técnica de aprendizaje, el uso de algún recurso de red, la asignación de ese recurso y un despliegue denso de BS, respectivamente. Posteriormente, los cuatro términos se concatenaron con el operador AND. La Tabla 2.3 muestra los términos clave y las palabras clave asociadas a ellos. Además, debido a que algunas bases de datos se enfocan en múltiples áreas de investigación, las búsquedas en las bases de datos *Springer Link* y *Scopus* se limitaron a las áreas de ingeniería e informática.

Tabla 2.3. Términos de búsqueda.

Término de búsqueda	Palabras clave
Método de aprendizaje	Learning OR artificial intelligence
Recurso	Resource OR power OR bandwidth OR channel OR spectrum
Gestión	Management OR allocation OR scheduling
Red ultra-densa	Ultra dense network OR UDN OR ultra-dense network OR small dense network OR dense network OR ultradense network

2.1.4 Criterios de inclusión

A partir de los artículos identificados por las búsquedas en las bases de datos, se aplicaron criterios específicos de selección para identificar los trabajos de investigación relevantes para esta revisión. Los criterios de inclusión se enumeran en la Tabla 2.4. Los criterios de inclusión 1 y 2 están relacionados con el objetivo de esta revisión. El criterio de inclusión 1 considera los artículos que se centran en la asignación de potencia y la asignación de ancho de banda en UDN, mientras que el criterio de inclusión 2 considera los trabajos que asignan estos recursos utilizando una técnica de ML. El criterio de inclusión 3 considera trabajos publicados en revistas de investigación, actas de congresos o capítulos de libros. Además, solo se consideraron los estudios escritos en inglés, como se indica en el criterio de inclusión 4. Por otro lado, se excluyeron aquellos trabajos de investigación sin información clara sobre el escenario de simulación o los detalles del modelo de ML implementado.

Tabla 2.4. Criterios de inclusión.

1. Estudios centrados en la asignación de potencia y ancho de banda en UDN
2. Estudios que implementan algunas técnicas de ML en la estrategia de asignación de recursos
3. Estudios publicados en revistas de investigación, revistas, actas de congresos o capítulos de libros
4. Estudios escritos solo en inglés

2.1.5 Proceso de selección

El proceso de selección es un proceso de filtrado que permite eliminar los trabajos menos relevantes de acuerdo con los criterios de inclusión. Primero, se eliminan los documentos duplicados. Después, se realiza un proceso de cribado con base al título y al resumen del documento siguiendo los criterios de inclusión mencionados en la Sección 2.1.4. Con los documentos seleccionados a partir del título y resumen, se realizó un segundo proceso de cribado con base en la lectura del texto completo. Los artículos seleccionados fueron aquellos que cumplen con los cuatro criterios de la Tabla 2.4. A este último conjunto de artículo se les aplicó el método de bola de nieve hacia adelante (forward snow-ball). El proceso de bola de nieve hacia arriba consiste en identificar aquellos artículos que citaron el trabajo de investigación en cuestión. Este último proceso se incluyó para considerar aquellos trabajos que no fueron detectados en la búsqueda de las bases de datos. Por último, se repitió el mismo proceso de selección a los documentos recuperados con el método de bola de nieve

hacia adelante, primero un cribado con base al título y resumen y, después, un segundo cribado con base a la lectura del texto completo. Todo el proceso de selección se llevó a cabo entre pares, se realizaron sesiones periódicas para discutir los resultados del proceso de selección y resolver desacuerdos.

2.1.6 Extracción de datos

Una vez terminado el proceso de selección, se extrajeron los siguientes datos: año de publicación, el problema de investigación atendido, el objetivo de optimización, la estrategia y el diseño del modelo de ML y el Indicador Clave de Rendimiento (KPI – key performance indicator) utilizado para evaluar la red celular. Dentro de la estrategia y diseño de la técnica de ML, se consideró la información del diseño del entorno del sistema inalámbrico, así como la generación de los datos utilizados para entrenar el modelo de ML, la entidad de red que ejecuta los algoritmos, las políticas de RA implementada y la estrategia para evaluar el modelo de ML.

2.2 Resultados

En esta sección, se describen los resultados obtenidos de la extracción de datos del conjunto de documentos seleccionados para realizar el análisis del trabajo relacionado. Además, se brinda una discusión de la información extraída derivando en las secciones de discusión, problemas abiertos y conclusiones.

2.2.1 Resultados de la búsqueda de las bases de datos

Las búsquedas en las bases de datos académicas se realizaron con fechas desde el 1 de enero de 2010 hasta el 31 de diciembre de 2022. La Figura 2.1 muestra el proceso de selección de los artículos considerados en esta revisión. Inicialmente se recuperaron 4,102 documentos (IEEE: 560, ACM: 605, Springer: 869, Scopus: 2,068). Se eliminaron, primero 259 documentos duplicados y después 3792 documentos cuando se les aplicó el primer y segundo cribado (título, resumen y texto completo). En total quedaron 51 trabajos de investigación. A estos 51 documentos se les aplicó el proceso de bola de nieve hacia delante, del que surgieron 143 documentos nuevos. Tras repetir los procesos de cribado (título, resumen y texto completo), solo se incluyeron tres documentos para su análisis. En total se

analizaron e incluyeron 54 documentos en esta revisión. Por otro lado, se eliminaron 88 artículos durante el proceso de cribado en texto completo debido a las siguientes razones: en 20 artículos de investigación no se implementó alguna técnica de ML en la estrategia de RA; en otros 20 trabajos de investigación no se centraron en problemas de RA sino en la descarga de tráfico, la asociación de UE (UEA – user equipment association) o la selección de BS en redes no-densas y UDN; en 36 artículos no consideraron el escenario de simulación UDN para resolver el problema de RA en su análisis; en 9 artículos los autores no describen el diseño experimental o detalles del algoritmo utilizado, como los híper-parámetros del modelo y la metodología de implementación; dos documentos estaban escritos en un idioma distinto al inglés; un artículo no fue accesible.

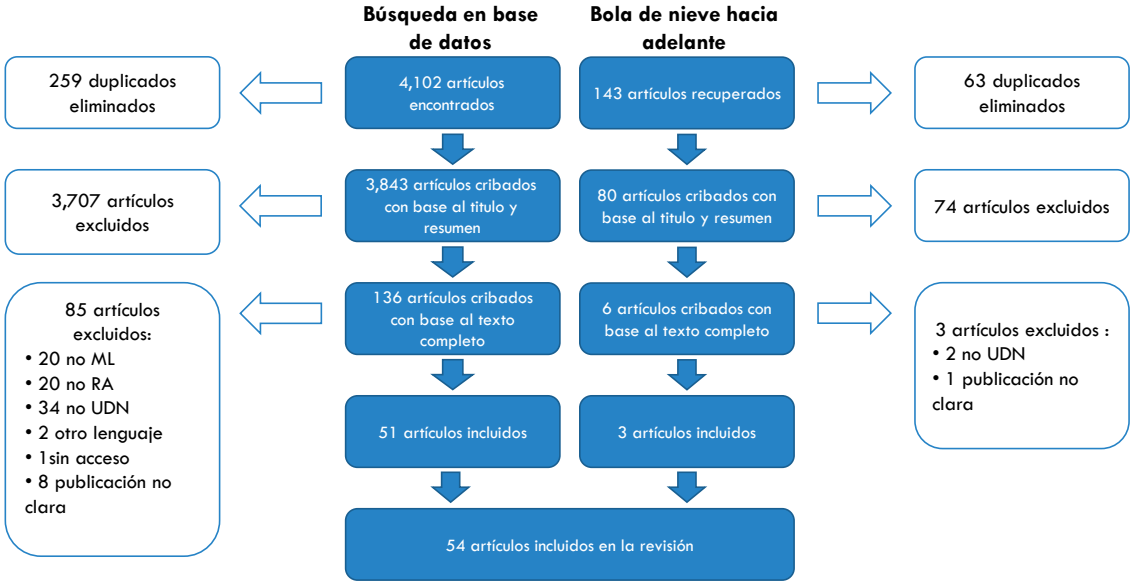

Figura 2.1. Proceso de selección de los artículos para la revisión sistemática.

2.2.2 Modelos de aprendizaje automático

De los 54 estudios reportados para resolver el problema de RA en B5G-UDN [11], [13-18], [39-85], se identificaron la implementación de 14 modelos de ML, los cuales se clasificaron en tres grupos principales de la siguiente manera (ver Figura 2.2).

- Grupo-1: Aprendizaje de refuerzo profundo (DLR): Actor-Critic Deep Learning (ACDL) [39], Bayes de Backprop Q-Network (BBQN) [64], Deep Deterministic Policy gradient (DDPG) [13], [65-69], [81-83], Deep Q-Network (DQN) [11],

[14-18], [40], [70-73], [84], Double Deep Q-Network (DDQN) [74], [75], Double Dueling Deep Q-Network(D3QN) [43] y Dueling Deep Q-Network (DuQN) [44].

- Grupo-2: Aprendizaje por refuerzo (RL): Q-Learning (QL) [48-63], [77-80], [85] y Multi-armed bandit (MAB) [46].
- Grupo-3: Redes Neuronales Artificiales (ANN): Long Short-Term Memory (LSTM) [45], Deep tree con Long-Short Term Memory (DLSTM) [41], Deep Neural Network (DNN) [42], Graph Neural Network (GNN) [76] y Neural Network (NN) [47].

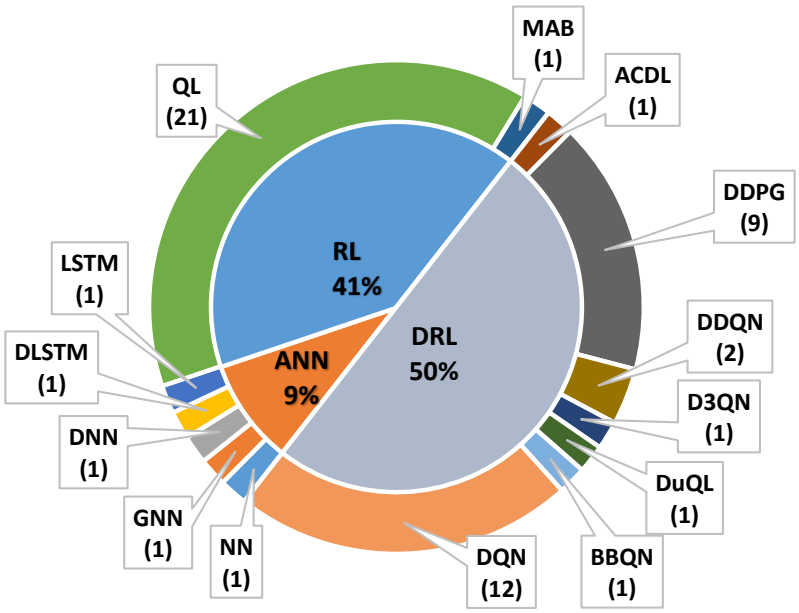

Figura 2.2. Distribución de los modelos de aprendizaje automático implementados para asignación de recursos en redes ultra-densas.

2.2.3 Distribución anual de los trabajos de investigación

Durante más de 20 años, los diseñadores de redes de comunicaciones inalámbricas han utilizado técnicas de ML para resolver problemas complejos como mejorar la retención de UE y maximizar la rentabilidad del operador [86]; eficientizar los protocolos de enrutamiento para redes de sensores submarinos [87]; controlar el tráfico inalámbrico en redes vehiculares [88]; o detectar la disponibilidad de espectro para Redes de Radio Cognitivas (CRN – cognitive radio networks) [89]. Sin embargo, como se muestra en la Figura 2.3, a partir del año 2016 inició la aplicación de técnicas de ML para solucionar el

problema de RA en UDN. En comparación con los modelos basados en ANN, existe una mayor tendencia por utilizar las técnicas de RL y DRL para resolver los problemas de RA. La razón es que las técnicas RL y DRL permiten que las entidades de la red y el UE móvil aprendan las condiciones de su entorno y después los controlen dinámicamente (p. ej., selección de canales y acceso al espectro) sin necesidad de contar con la información precisa de las condiciones de la red inalámbrica tal y como lo requieren los modelos ANN [90].

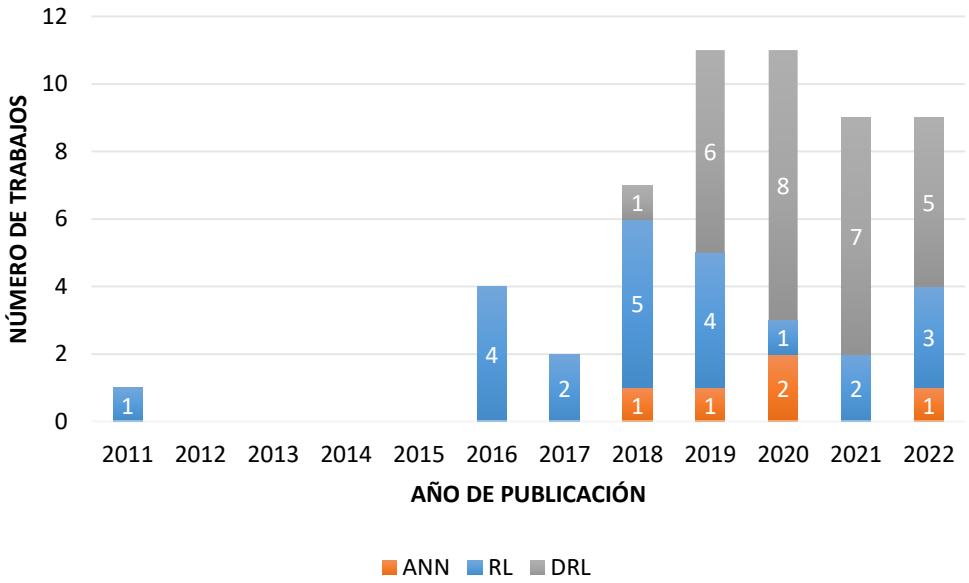

Figura 2.3. Distribución anual de los estudios seleccionados que implementan modelos de aprendizaje automático para la asignación de recursos en redes ultra-densas.

2.3 Técnicas de aprendizaje automático implementadas en estrategias de asignación de recursos

En esta sección se describe el esquema general del proceso de aprendizaje (ver la Figura 2.4) y las características de los diferentes modelos de ML para resolver el problema de RA en UDN. El modelo ML aprende al observar la información de la red y recibir retroalimentación sobre el rendimiento de la estrategia de RA actual. Con base a la información recibida, el modelo de ML actualiza sus parámetros para mejorar la estrategia de RA. Este proceso se repite hasta lograr una estrategia de toma de decisiones óptima. Con base a los grupos de ML mencionados en la subsección 2.2.2, esta sección se dividió en las subsecciones; modelos basados en ANN, basados en RL y basados en DRL.

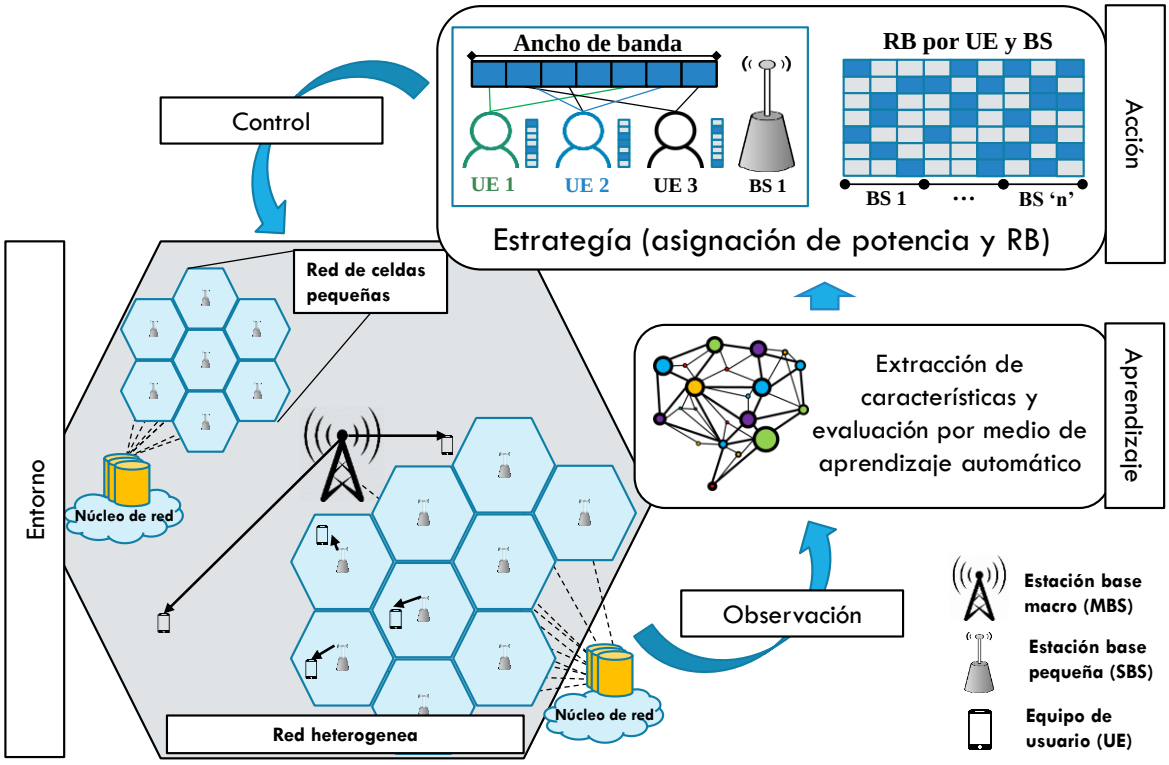

Figura 2.4. Esquema general de la implementación de aprendizaje máquina para la asignación de recursos en redes ultra-densas.

2.3.1 Modelos basados en redes neuronales artificiales

Los algoritmos basados en ANN aprenden a realizar tareas como la RA sin conocer las condiciones del entorno de implementación. El modelo de Red Neuronal (NN – neural networks) está formado por la capa de entrada, la capa oculta y la de salida. Los algoritmos de NN se entrenan a partir de un conjunto de datos expertos. Es decir, aquellos datos que contienen la información del entorno y soluciones al problema de RA resuelto por otros algoritmos de optimización. Este conjunto de datos los utiliza el modelo NN para optimizar sus parámetros y mejorar la precisión de los valores predichos (valores de la capa de salida). Los parámetros se ajustan según al error que exista entre los valores predichos por la NN y los valores del conjunto de datos experto [91]. Asimismo, la red NN se vuelve DNN al considerar múltiples capas ocultas entre las capas de entrada y salida. El flujo de datos desde la capa de entrada a la de salida crea un mapa de neuronas virtuales y a cada conexión entre neuronas se le asigna un peso. Estos pesos se ajustan en cada paso del proceso de optimización cuando el modelo NN no reconoce un patrón en particular. Además, se pueden diseñar diferentes arquitecturas de NN para varios propósitos. Por ejemplo, una NN

recurrente como la Long Short-Term Memory (LSTM) [92] permite conectar neuronas de capas anteriores, lo cual es de utilidad para resolver problemas de predicción en secuencia. Mientras tanto, arquitecturas como las redes neuronales convolucionales (CNN) [93] y Graph Neural Network (GNN) [94], se implementan para procesar datos representados como imágenes o gráficos.

Los modelos basados en ANN se han utilizado para resolver problemas de RA con el propósito de mejorar la eficiencia energética (EE) [42], la capacidad [47], [76] y el control de tráfico [41], [45] en redes UDN. La Tabla 2.5 muestra las características de diseño, optimizador, la capa de activación y la estrategia de entrenamiento, utilizadas por estos modelos de RA basados en ANN.

En [47], se utiliza una NN para minimizar la interferencia en una red móvil celular LTE durante la asignación de canales. Los autores implementan un modelo Min k-Cut modificado y algoritmos de gráfico de conflicto. La NN extrae la relación de interferencia de los usuarios de la red respecto a cada UE. El conjunto de datos consiste de los UE que están utilizando el mismo bloque de recurso (RB – resource block) en el mismo intervalo de tiempo, mientras que la etiqueta es definida como la relación señal a interferencia más ruido

Tabla 2.5. Características de diseño de modelos basados en ANN para asignación de recursos en UDN. ReLU: Unidad lineal rectificada. GA: Gradiente ascendente. PA: Asignación de potencia. UEA: Asociación de usuarios. SGD: Gradiente descendiente estocástico. DTM: Deep Tree Model.

Objetivo	Referencia	Modelo	Entrenamiento	Optimizador	Función de activación
Maximización de EE	42	DNN	El conjunto de datos se generó resolviendo un algoritmo de gradiente iterativo. Los conjuntos de datos generados para el entrenamiento y la validación fueron 15000 y 1000, respectivamente	Adam	ReLU
Maximización de capacidad	47	NN	El conjunto de datos contiene miles de millones de muestras generadas a partir de la plataforma LTE-Sim, cada muestra consta del UE interferente potencial y la SINR de su enlace ascendente	-	-
	76	GNN	Primero se entrenó el modelo con varias soluciones de optimización para lograr generalización. Luego, el modelo se ajusta con los datos del escenario objetivo	GA	PA: Sigmoid UEA. Softmax
Control de tráfico	41	DLSTM	El entrenamiento se realizó con diferentes relaciones fijas de enlace ascendente/descendente. Además, se utilizó un modelo profundo con estructura de árbol para mejorar la regularización y reducir la complejidad del modelo LTSM	SGD	DTM: ReLU LSTM: Sigmoid and tanh
	45	LSTM	El entrenamiento consiste en una secuencia de paquetes de datos en el buffer de envío con relaciones fijas de enlace ascendente/descendente. Después, el modelo se entrena cada vez que se detecta un cambio de relación	-	Sigmoid and tanh

(SINR – signal-to-interference-noise-ratio) del enlace ascendente. El conjunto de datos utilizado para el entrenamiento se recopiló de una plataforma de simulación para sistemas LTE llamada LTE-Sim. En [76], se propone una GNN para resolver el problema de Asociación de Usuarios (UEA) a la radio base y la asignación de potencia de transmisión. La UEA se realiza cuando se descarga de tráfico móvil a una BS, esto se logra desconectando a uno o varios UE. Por lo que se debe buscar un nuevo enlace de comunicación para conectar/asociar al UE con una nueva BS. La GNN se utiliza para extraer la información de CSI, matriz de asociación y potencia de transmisión de los nodos. Luego, se propone un esquema de entrenamiento de dos fases combinando el aprendizaje supervisado (i.e., usa base de datos con etiquetas) y no supervisado (i.e., usa base de datos sin etiquetas). Primero, el modelo explota la habilidad de generalización durante el primer entrenamiento en un entorno de simulación considerando diversas condiciones de la red, mientras que en la segunda fase, se mejora el rendimiento del modelo con el entrenamiento en línea al adaptarse a los escenarios específicos en tiempo real. Los resultados muestran que el uso de la técnica GNN logra un mayor rendimiento y convergencia que los modelos de NN y CNN. Además, su propuesta supera a las técnicas tradicionales, como el método de asociación de tasa máxima alcanzable con potencia máxima (MARAMP – maximal achievable rate association with maximum power) y el método de asociación de utilidad de suma máxima con potencia máxima (MSUAMP – maximual sum-utility association with máximum power). En [42], se propone una red neuronal profunda (DNN) centralizada para asignar el nivel de potencia de transmisión a cada enlace del UE en la UDN. La DNN se entrena con los datos obtenidos por un método de gradiente iterativo, y luego, los datos se normalizan y se formatean con distribución normal con media cero y regularización L2, respectivamente. Además, proponen una DNN distribuida para una red UDN de gran escala. La DNN distribuida divide la DNN centralizada en varios modelos de DNN que se entrenan en paralelo. Todos los parámetros se recopilan por un controlador central que actualiza todos los parámetros de las DNN. Este procedimiento reduce considerablemente el tiempo de entrenamiento y hace que el sistema sea robusto debido el entrenamiento con base a diferentes redes de pequeña escala. Los resultados muestran una precisión entre el 97 y el 98.4%, casi diez veces menos tiempo de operación con una ligera diferencia en EE con respecto al algoritmo de gradiente iterativo utilizado para el entrenamiento. En [41] y [45], el control del tráfico se atiende mediante la

implementación del modelo LSTM, que utiliza los datos recopilados al instante y los datos almacenados en instantes previos, para determinar los valores de predicción. Los modelos LSTM se utilizan para cambiar la relación de enlace ascendente y enlace descendente antes de que ocurra una congestión en un sistema con transmisiones dúplex por división de tiempo. A diferencia de [45], en [41], los autores implementaron un modelo profundo basado en árboles de decisión, antes de enviar la información al modelo LSTM para reducir los parámetros de los datos recopilados en el dominio espacial. Ambos trabajos ofrecen una mejora en el rendimiento de la red con respecto a los métodos en los que las políticas de RA de la red cambian una vez que se produce una congestión de tráfico en la red.

2.3.2 Modelos basados en el aprendizaje por refuerzo

Los modelos basados en Aprendizaje por reforzamiento (RL) tienen la capacidad de asignar los recursos de la red dinámicamente. Esta asignación de recursos es posible al extraer el conocimiento de una gran cantidad de datos sin necesidad de modelos matemáticos explícitos. Es decir, a diferencia de los modelos basados en ANN, los modelos basados en RL aprenden de interactuar con el entorno de manera secuencial, lo que les permite adaptarse al entorno en el que son implementados. Sin embargo, en ausencia de un conjunto de datos para su entrenamiento, los modelos basados en RL aprenden de sus experiencias [20]. Estas experiencias se generan mediante la interacción directa con el entorno de implementación. De esta forma, los modelos basados en RL toman acciones con base a sus experiencias pasadas. Estas acciones se recompensan positiva o negativamente de acuerdo con el efecto de la acción en el entorno de implementación. Al finalizar la interacción, la información de la acción tomada, la información del estado y el valor de recompensa se almacenan en una matriz llamada tabla-Q [95]. El proceso de aprendizaje consiste en ejecutar acciones de acuerdo con una estrategia de exploración, utilizando el resultado de cada acción ejecutada para actualizar los valores de la tabla-Q. Los valores en la tabla-Q indican cual fue la mejor acción con respecto a la información del entorno. Este proceso de actualizar los valores de la tabla-Q se repite hasta que se cumple un criterio final (e.g., el número de episodios). La Tabla 2.6 muestra las características de diseño de los trabajos que aplican modelos basados en RL para resolver el problema de RA en UDN. Estos trabajos de investigación se agruparon, según el objetivo de optimización, en minimización de retardo[63], maximización de la

eficiencia energética (EE) [50], [54], [80], [85], mitigación de interferencia, incluyendo el control de interferencia y coordinación de interferencias entre celdas (ICIC) [46], [51], [52], [55], [57], maximización de capacidad [48], [49], [53], [58-62], [77-79] y maximización de utilidad [56].

En [63], se propone un modelo de RA para minimizar el retardo de los paquetes de datos transmitidos en un sistema LTE. El objetivo es reducir la latencia y aumentar la equidad entre UE móviles y dispositivos MicroGrid (MGD). Cada BS/MBS asigna los RB de acuerdo con la función de recompensa, que se adapta con un peso escalar para controlar las prioridades entre los tipos de tráfico en cada tipo de dispositivo logrando un balance entre UE y MGD. La conclusión de este trabajo es que entre mayor sea el espacio de acción mejores acciones de RA se pueden aprender, pero a costa de un mayor retardo.

Tabla 2.6. Características de diseño de las técnicas basadas en RL para asignación de recursos en UDN. ST: Transmisor secundario (i.e. PBS, FBS and D2D). RBG: Grupo de Bloque de recursos. CQI: Indicador de Calidad del Canal. D2D: Dispositivo a Dispositivo. SE: Eficiencia Espectral. Los números arábigos indican la implementación de más de un modelo.

Objetivo	Ref	Modelo	Estructura del modelo				
			Agente	Acción	Estado	Recompensa	Exploración
Minimización de retardo	63	QL	eNB/SBS	Conjunto de RB a sus UE	CSI de UE y SBS	Retardo considerando MGD y UE	ε-greedy
Maximización de EE	50	QL	SUE	Asignación de potencia y RB	RB asignado y nivel de potencia del SUE	EE considerando un umbral SINR de los UE	Distribución de Boltzmann
	54	QL	ST	Asignación de potencia	Identificador de ST y potencia de transmisión	EE	Distribución de Boltzmann
	80	QL	SBS	Niveles de potencia	Potencias máxima, mínima y nivel de la SBS	EE considerando el número de UE en interrupción	ε-greedy
	85	QL	Cabeza del clúster de las FBS	Asignación de potencia y RB	Retardo, SINR, tasa de datos requeridos y asociación de los UE	EE considerando la QoS de los UE	Distribución de Boltzmann
Mitigación de interferencia	46	MAB	PBS	Porción de ancho de banda	Índice de celda y banda espectral	Capacidad promedio	Función de decisión
	51	QL	1: FAP 2: FAP 3: FAP	1: Potencia de transmisión 2: Igual a 1 3: Igual a 1	1: Capacidad del MUE 2: Ubicación de la MBS y el MUE respecto a la FAP 3: Igual a 2	1: Métrica considerando la ubicación y QoS del MUE 2: Métrica considerando la QoS del MUE y la capacidad del FUE 3: Igual a 2	1: ε-greedy 2: ε-greedy 3: ε-greedy
	52	QL	SBS	Selección del RB	-	Capacidad	ε-greedy
	55	QL	SBS	Asignación de potencia	Densidad de UE y SINR previó del UE	Métrica considerando la densidad de UE, SINR y potencia de transmisión	ε-greedy
	57	QL	SBS	Asignación de potencia	Interferencia máxima y SINR de la SBS	Métrica considerando capacidad e interferencia	-
Maximización de capacidad	48	QL	eNB y SBS	Asignación de RBG	CQI y tasa de paquetes enviados	Capacidad	ε-greedy
	49	QL	SBS	Asignación de potencia	Capacidad del UE en la celda agresora	Capacidad de la SBS	ε-greedy
	53	QL	FBS	Asignación de potencia	Capacidad de los FUE y MUE, interferencia hacia las FBS y distancia de las FBS a la MBS	Capacidad considerando los requerimientos mínimos de los FUE y MUE	ε-greedy
	58	QL	1: Cabeza de clúster 2: MBS	1: Asignación de RB dentro del clúster	1: QoS de la MBS 2: Igual a 1	1: Capacidad del clúster por RB	ε-greedy

Objetivo	Ref	Modelo	Estructura del modelo				
			Agente	Acción	Estado	Recompensa	Exploración
				2: Asignación de RB a cada clúster		2: Capacidad promedio de los clúster	
	59	QL	SBS	Asignación de potencia	Distancia concéntrica de la cabeza del clúster a los UE	Capacidad considerando la QoS de los UE asociados a las SBS	ε-greedy
	60	QL	SBS	Asignación de potencia	Potencia de transmisión de los vecinos y SINR objetivo de los MUE	Capacidad priorizando la QoS de los UE	Distribución de Boltzmann
	61	QL	FBS	Asignación de potencia	Distancia de los FBS en la vecindad respecto a los MBS y MUE	Métrica priorizando la capacidad de los FUE considerando la QoS de los FUE y MUE	ε-greedy
	62	QL	FBS /D2D	Asignación de potencia, RB y modulación del transmisor secundario	Identificador del transmisor, RB disponibles y SINR de los transmisores vecinos	Capacidad de los MUE y FUE considerando la SE del receptor secundario	ε-greedy
	77	QL	SBS	Asignación de potencia	Interferencia máxima de la SBS e interferencia del clúster	Capacidad priorizando la SBS y considerando los usuarios sin servicio	ε-greedy
	78	QL	SBS	Asignación de potencia	Distancias entre la SBS a la MBS y la SBS al MUE	Capacidad del UE considerando el SINR mínimo del MUE y SUE	ε-greedy
	79	QL	SBS	Asignación de potencia	Distancia concéntrica entre la SBS a la MBS y el SBS al MUE	Capacidad del UE considerando el SINR mínimo del MUE y SUE	ε-greedy
Maximización de utilidad	56	QL	UE	Asignación de potencia y UEA	SINR, estado de asociación y potencia del UE	Métrica considerando la UEA, EE y QoS	Distribución de Boltzmann

Los trabajos [50], [54], [80] y [85] abordan el problema de maximizar la EE. En [50], los autores proponen un modelo QL activado por eventos con el fin de ahorrar recursos computacionales. El equipo de usuario de estación base pequeña (SUE) actúa si la diferencia entre la recompensa del agente y las recompensas promedio de los SUE que usan el mismo canal es mayor a un valor de umbral. Este enfoque activado por eventos logra una mejor EE que el modelo QL que se activa cada cierto intervalo de tiempo. En [54], los autores utilizan un esquema de aprendizaje intuitivo para inferir la información local de otros transmisores secundarios (ST) mediante sus interacciones con el entorno y sus experiencias pasadas. Los ST considerados son estaciones base Pico (PBS), estaciones base Femto (FBS) y dispositivo a dispositivo (D2D). Además, se propone una función de valor-Q para reducir el espacio de estados, dando como resultado un mejor rendimiento que los algoritmos convencionales para maximizar la EE garantizando los niveles de QoS a los UE primarios y secundarios. En [80], se propone un algoritmo de control de potencia basado en el modelo QL distribuido para maximizar la EE y minimizar el número de interrupciones (outage) del UE debido a la interferencia. En este esquema, el UE considera solo su acción pasada para la toma de decisiones, mientras que la función de recompensa considera el rendimiento global de la red. Los resultados muestran que la complejidad computacional se reduce en comparación cuando se utiliza el modelo QL centralizado, en el que la tabla-Q aumenta exponencialmente con el

número de agentes, incrementándose, como consecuencia, el número de acciones y de estados. Además, se logra una tasa de convergencia más alta con su modelo QL distribuido considerando una recompensa independiente, evaluada en escenarios de distribución de tráfico espacial de UE uniformes y no uniformes. En [85], se maximiza la EE en una red ultra-densa formada por femto-celdas. El modelo considera la QoS de los usuarios femto-celulares (FUE) para asignar los recursos. La estrategia agrupa las femto-radio bases (FBS) utilizando el algoritmo de agrupamiento K-medias (K-means), este considera la ubicación geográfica de las FBS y su carga de tráfico en cada clúster. Luego, un algoritmo QL asigna los RB a cada FBS considerada como la cabeza del clúster. Este modelo QL se entrena de forma cooperativa utilizando la información de otros agentes que se encuentra almacenada en un servidor en la nube. Esta propuesta presenta mejores resultados de la EE, capacidad y velocidad de convergencia que el modelo QL independiente y el método Stackelberg.

La aplicación de técnicas RL para el control de interferencia se describen en los trabajos [46], [51], [52], [55] y [57]. En [46], se implementa un algoritmo Multi-armed bandit (MAB) para seleccionar la porción del ancho de banda que mejor controla la interferencia entre celdas (ICIC). El algoritmo MAB accede secuencialmente a todas las bandas de espectro disponibles y selecciona el grupo de canales que ofrece la mayor recompensa. Además, se propone una función de recompensa para seleccionar la porción de banda espectral para transmitir. El valor de recompensa de cada banda espectral considera su aportación en el rendimiento en la red y el número de veces que la banda espectral es elegida consecutivamente. Esta última consideración se implementa para evitar un comportamiento egoísta que seleccione siempre la banda espectral con mayor recompensa y evite la exploración del resto de las bandas espectrales. Los resultados muestran un mayor rendimiento cuando se utilizan porciones de espectro dinámicamente en lugar de los esquemas basados en partición del espectro que dividen el espectro para reutilizarlo en cada celda y lo mantienen las porciones de banda espectral fijas. En [51], los autores evalúan su propuesta en una HetNet formada por un conjunto de FBS desplegadas sobre una red macrocelular con una macro estación base (MBS). En este trabajo se proponen tres variantes del modelo QL para la asignación de potencia; distribuido, formulado y cooperativo. El esquema distribuido propone mejorar la capacidad del FUE mientras mantiene la QoS del usuario de la macrocelda (MUE). Por otra parte, el algoritmo formulado, modifica su estado,

considerando la ubicación de la MBS, el MUE y el FBS. Mientras tanto, el esquema QL cooperativo reduce la complejidad computacional permitiendo que los agentes experimentados (es decir, los agentes ya entrenados) compartan información con nuevos agentes que comparten un estado similar con el fin de acelerar su convergencia en el entrenamiento. Los resultados muestran que la ubicación del MUE es un factor decisivo para cumplir con los requisitos de QoS de la red.

En el trabajo [52], los autores proponen una arquitectura de doble salto, esto es, conexiones a la red de acceso y red de auto-retorno compartiendo el mismo espectro. La SBS está a cargo de las transmisiones de la Hub Base Station (HBS) a la SBS y de la SBS al UE. Para reducir la complejidad computacional del modelo QL, los autores consideran un modelo QL de un solo estado, simplificando los pares acción-estado a un formato sin estado. En la inicialización, los agentes eliminan el canal más interferente del conjunto de canales disponibles. Luego, los agentes asignan un valor-Q para cada acción, que guía su toma de decisiones. Como resultado, la capacidad del enlace de la red de acceso y la red de auto-retorno aumentan y el tiempo de convergencia se reduce comparado con un enfoque de radio cognitivo. El trabajo en [55] propone un modelo QL con un método Transferencia de Aprendizaje (TL – transfer learning) para acelerar la velocidad de aprendizaje en una red de celdas pequeñas (SCN – small cell network). Al igual que en [51], la tabla-Q se actualiza con información de nuevos agentes a partir de los valores-Q de agentes entrenados. Sin embargo, la recompensa en [55] se basa en la densidad del UE, la SINR y la potencia de transmisión, centrándose en reducir la interferencia entre las celdas y en ahorrar el consumo de energía de las BS. Por otra parte, en [57] se implementan la estrategia de gráfico de conflicto para el agrupamiento de BS y el modelo QL para la gestión de interferencia. Los agentes asignan la potencia de transmisión a los diferentes RB de acuerdo con la interferencia que provocan el resto de los agentes y a la SINR general de la red. Los resultados muestran una mejora en el rendimiento de la red. Sin embargo, la capacidad máxima alcanzable se reduce a expensas de una mejor capacidad en los UE localizados en los bordes de las celdas.

Por otra parte, en [48], [49], [53], [58-62] y [77-79], se atiende específicamente la maximización de la capacidad de la red UDN. Los autores en [48] consideran un escenario en el que se presenta la coexistencia entre dispositivos de uso intensivo de datos (DID) y UE tradicionales que transmiten solo voz y video. Los DID son aplicaciones emergentes que se

espera sean frecuentes en las futuras redes móviles, como la realidad aumentada, la realidad virtual y las aplicaciones táctiles. La red cuenta con múltiples agentes, en la que la MBS y las SBS se encargan de la asignación de RB a sus usuarios adjuntos, respectivamente. Para reducir el espacio de acción, se asignan grupos de bloque de RB continuos (RGB – resource group block) en lugar de RB individuales. Los resultados de capacidad, retardo y equidad de esta propuesta superan a los obtenidos con el algoritmo de equidad proporcional (PF – proportional fair) para diferentes densidades de DID. El trabajo en [49] evita la interferencia entre SBS mediante la implementación de un esquema de Coordinación de Interferencia Entre Celdas (ICIC - inter-cell interference coordination) que adapta las tramas de datos bajo el esquema de apagado temporal de subtramas casi vacías (ABS - almost blank subframes) y un modelo QL para el control de potencia. Las subtramas envían únicamente información de señalización y no de información de datos. El esquema ICIC propuesto adapta la relación ABS de acuerdo con la carga de la celda más interferente. Este esquema permite que las celdas (llamadas celdas agresoras) transmitan con poca potencia (controlada por el modelo QL) en las subtramas dedicadas a la señalización. Los resultados muestran que el control dinámico de potencia del esquema ABS de baja potencia supera al esquema ABS convencional a medida que aumenta la densidad de los UE en la red. El trabajo en [53] considera un modelo QL distribuido para el control de potencia en una Red Auto-Organizada (SON – self-organizing network). A medida que la densificación del sistema aumenta, los autores consideran dos esquemas de entrenamiento para los nuevos agentes, es decir, Aprendizaje Independiente (IL – independent learning) y Aprendizaje Cooperativo (CL – cooperative learning). Para resolver el problema de optimización, diseñan una función de recompensa para cumplir con los requisitos de QoS de los FUE y MUE. Los resultados muestran que el IL logra en los FB mayor capacidad y mayor consumo de energía, mientras que, el CL logra mayor capacidad en la MBS y menor consumo de energía a costa de mayor tráfico de señalización.

Por otra parte, los autores en [58] consideran dos enfoques de RA; el enfoque centralizado, donde la MBS asigna bloques de recursos a cada líder de clúster y; el enfoque distribuido, donde el líder de cada clúster asigna los RB a cada uno de las BS dentro de su clúster. Los resultados muestran que la red logra un mejor desempeño con el enfoque centralizado. Sin embargo, el amplio espacio de acción del esquema centralizado requiere un

mayor tiempo de convergencia comparado con el enfoque distribuido. Los autores en [59] proponen un modelo QL para formar el clúster y asignar la potencia. El estado se define con base a la distancia concéntrica a partir de la SBS considerada como la cabeza del clúster seccionado por zonas. Además, se propone una recompensa para satisfacer la QoS y proporcionar equidad entre los miembros del clúster a medida que el tamaño del clúster incrementa. En [60], los autores dividen el área de cobertura de la MBS en dos zonas, el borde de la celda y el centro de la celda. Estas se definen como las zonas de baja y alta interferencia, respectivamente. El modelo QL se implementa en las SBS de la región del borde de la celda para asignar la potencia de transmisión a sus UE con el fin de maximizar el rendimiento y al mismo tiempo que garantiza la QoS del MUE. El trabajo en [61] implementa el modelo QL para proporcionar equidad en todo el sistema de red. Los agentes presentan dos modos de operación, IL y CL (al igual que en [53]). Con el IL, los agentes aprenden a través de la interacción con el entorno, mientras que, con CL, los agentes aprenden de los agentes experimentados. La complejidad del algoritmo se reduce utilizando el enfoque cooperativo en comparación con el aprendizaje individual, ya que los agentes pueden compartir sus experiencias en lugar de descubrir toda la información del entorno (es decir, realizar acciones de exploración) por sí mismos. En [62], los agentes (i.e., FBS y D2D) trabajan de forma cooperativa para compartir su información del estado. El agente puede asignar RB, potencia y adaptar la modulación de la transmisión. Además, se implementaron dos mecanismos para el modelo QL. El primer mecanismo reduce la tasa de exploración a medida que el entrenamiento avanza. El segundo mecanismo modifica la tasa de aprendizaje (híper-parámetro del modelo QL) con el objetivo de aprender más rápido cuando el valor-Q decae y más lento cuando el valor-Q incrementa. Estas modificaciones evitan que el mecanismo de aprendizaje dependa únicamente de las métricas de rendimiento. Los resultados muestran un mejor desempeño en capacidad, SE y equidad. Los autores en [77] implementan un modelo QL para controlar la potencia. Primero, introducen un análisis de la interferencia de la red mediante la teoría de grafos. Luego, la información recopilada del análisis de la interferencia se utiliza en el estado del modelo QL. Los resultados muestran que esta propuesta describe mejor la interferencia de toda la red, logrando una mayor capacidad que otros algoritmos como el algoritmo de llenado de agua (water filling algorithm). En [78] y [79], se implementa el modelo QL para maximizar el rendimiento de

los MUE y SUE. Los autores implementan un control de potencia adaptable en la SBS, asumiendo funcionalidades SON para la auto-optimización. Para la validación, consideran diferentes escenarios para mitigar la interferencia entre redes (por ejemplo, entre redes de MBS y SBS). La función de recompensa considera los requisitos mínimos de QoS de los UE. Por otra parte, en [79] se propone una modelo QL cooperativo. Este esquema comparte la información de la tabla-Q entre los agentes cercanos durante el proceso de aprendizaje. Los resultados muestran que el esquema de cooperación logra una mayor capacidad en los SUE a medida que la densidad de SUE aumenta comparado con el esquema de aprendizaje independiente.

Por último, algunos trabajos proponen sus KPI para maximizar el rendimiento de la red, denominados como maximización de utilidad. La función de utilidad consiste en considerar el rendimiento de diferentes KPI. Por ejemplo, la función de utilidad de los autores en [56] considera la EE, el balanceo de cargas a través de la asignación de potencia y la UEA. Para maximizar la utilidad proponen un modelo QL que logra un equilibrio de cargas que beneficia al número de UE asociados a la SBS, lo que logra un alto desempeño en la EE a medida que la red se densifica.

2.3.3 Modelos basados en el aprendizaje por refuerzo profundo

Los algoritmos basados en Aprendizaje por Refuerzo Profundo (DRL) tienen la misma estructura que los basados en RL. Sin embargo, en este caso, el valor-Q se aproxima a través de una DNN [38]. La DNN se vuelve necesaria en problemas de alta dimensionalidad debido que a medida que el tamaño de la red incrementa, también lo hace el espacio de acción y de estado, lo que dificulta encontrar políticas óptimas (es decir, una acción que obtenga el máximo beneficio a largo plazo). La capacidad de generalización de la DNN permite aproximar la función de valor-Q eficientemente, en lugar de actualizar la tabla-Q, la cual requiere evaluar, cientos de veces, todas las acciones para ajustar el valor-Q de cada elemento de la matriz. Mayor detalle del funcionamiento de los modelos basados en DRL se describe en el capítulo 3. La Tabla 2.7 muestra las características de diseño de los trabajos que utilizan modelos basados en DRL para resolver los problemas de asignación de recursos en UDN. Estos modelos se agruparon de acuerdo con los siguientes objetivos: minimización del Costo Computacional (CC) [65], maximización de EE y eficiencia espectral (SE – spectral

eficiency) [44], [75], minimización del consumo de energía (EC – energy consumption) [67], maximización de EE [13], [14], [17], [71], [73], [74], mitigación de interferencia [15], maximización conjunta de SE, EE y equidad [16], maximización de SE [11], maximización de EE y capacidad [66], maximización de capacidad [18], [64], [68-70], [72], [81-84], maximización de satisfacción del UE [39], [40] y maximización de utilidad [43].

Tabla 2.7. Características de diseño de las técnicas basadas en DRL para asignación de recursos en UDN.

Objetivo	Ref	Modelo	Estructura del modelo				
			Agente	**Acción**	**Estado**	**Recompensa**	**Exploración**
Minimización de CC	65	DDPG	SBS	Potencia de transmisión y descarga de tareas	Ganancia del canal y SINR de cada UE	Costo computacional	Ruido de exploración
Maximización de EE y SE	44	DDQN	MgNB	Asignación de RB	Identificador, capacidad y asignación de RB de las SC	EE y SE	ε-greedy
	75	DDQN	MBS	Asignación de RB	Asignación de RB y capacidad	EE y SE	ε-greedy
Minimización de EC	67	DDPG	SBS	Asignación de potencia, RB, recursos computacionales y descarga de tráfico	Ganancia del canal, interferencia y perfiles de tareas de los UE de la SBS	Métrica considerando el consumo de energía y los requerimientos de latencia de las tareas de los UE	-
Maximización de EE	13	DDPG	Entidad central	Asignación de potencia	Recolección de energía, nivel de batería, carga de tráfico y capacidad de todas las SBS	EE	Ruido de exploración
	14	DQN	UE	Asignación de potencia y UEA	Asignación de potencia y UEA	EE de todos los UE	ε-greedy
	18	DQN	AP central	Asignación de potencia	Capacidad y potencia de transmisión	EE	ε-greedy
	71	DQN	Entidad central	Asignación de potencia y UEA	Volumen del tráfico, estado del canal y potencia de transmisión	EE	ε-greedy
	73	DQN	AP central	Asignación de potencia y RB	Capacidad y potencia de transmisión de los UE	Capacidad	ε-greedy
	74	DDQN	SBS	Asignación de potencia	Interferencia de los UE y capacidad de la SBS	Capacidad y EE del sistema	ε-greedy
Mitigación de interferencia	15	DQN	SBS	Potencia de transmisión del UE	SINR, estado del canal del UE y densidad de UE estimada	Métrica considerando capacidad, consumo de energía e interferencia	ε-greedy
Maximización de EE, SE y equidad	17	DQN	SBS	Asignación de potencia y RB	Interferencia y UEA	Métrica considerando la EE y la varianza de la capacidad entre los UE	Generado por una DNN
Maximización de SE	39	DQN	SBS	Asignación de potencia	Potencia de transmisión, capacidad y SINR de las SBS cercanas	SE en el enlace	ε-greedy
Maximización de EE y capacidad	66	DDPG	Entidad central	Asignación de potencia, UEA y potencia disponible de la BS	Potencia de transmisión, tiempo de transmisión y capacidad de los UE	EE y capacidad	Ruido de exploración
Maximización de capacidad	40	DQN	1: FBS 2: MBS	1: Potencia de transmisión del FUE 2: Potencia de transmisión del MUE	1: Distancia del FBS al MBS y FBS al MUE 2: Distancia entre el MUE al MBS y FBS al MBS	1: Capacidad considerando la QoS del FUE 2: Capacidad considerando la QoS del MUE	1: ε-greedy 2: ε-greedy
	64	BBQN	SBS	Asignación de potencia	CSI, potencia de transmisión y capacidad	Capacidad del sistema	BNN

Objetivo	Ref	Modelo	Estructura del modelo				
			Agente	**Acción**	**Estado**	**Recompensa**	**Exploración**
	68	DDPG	SBS	Asignación de potencia	Ganancia del canal e interferencia	Capacidad	Ruido de exploración
	69	DDPG	1: Entidad central 2: SBS	1: Potencia de transmisión y transferencia de energía de cada SBS 2: Potencia de transmisión y transferencia de energía en su SBS	1: SINR, potencia de batería, y energía recolectada de las SBS 2: SINR, potencia de batería, y energía recolectada de los UE en su SBS	1: Capacidad de la red 2: Capacidad de la SBS	Ruido de exploración
	70	DQN	UE	Asignación de RB y UEA	Capacidad y estado de requerimiento del UE	Métrica considerando la capacidad y la QoS de los UE	ε-greedy
	72	DQN	SBS	Asignación de potencia	Tiempo promedio de los paquetes y estado de carga de la SBS	Capacidad considerando la potencia mínima requerida	ε-greedy
	81	DDPG	MBS	Asignación de potencia	CSI y matrices de cooperación y de potencia asignada	Capacidad del sistema	Ruido de exploración
	82	DDPG	Entidad central	Asignación de potencia y RB	Ubicación de UE, satélites e intensidad de lluvia	Capacidad de los UE	Entropía
	83	DDPG	Entidad central	Asignación de potencia y encendido/apagado	Tasa de UE atendidos	Métrica considerando la capacidad penalizada por el consumo de energía	Ruido de exploración
	84	DQN	gNB	Asignación de RBG	CSI, capacidad mínima, RBG asignado a cada UE	Capacidad del sistema	ε-greedy
Maximización de la satisfacción del UE	11	ACDL	Entidad central	Asignación de RB y UEA	Capacidad y nivel de energía de la batería de la BS	Métrica considerando el RB asignado y el RB requerido	ε-greedy
	16	DQN	SBS	Asignación de RB	Parámetros del entorno, demanda de QoS y QoS proveído	Satisfacción de UE	ε-greedy con mecanismo heurístico
Maximización de utilidad	43	D3QN	UE	Asignación de RB y UEA	Estado de la demanda de todos los UE	Métrica considerando la diferencia entre la ganancia y el costo de transmisión	ε-greedy

En [65], la minimización de CC se aborda para un sistema de Computación en el Extremo Móvil (MEC - mobile edge computing) de Acceso Múltiple No Ortogonal (NOMA – non-orthogonal multiple access). Para atender el problema, los autores proponen el algoritmo de optimización iterativo de User Cluster Matching (UCM) en conjunto con Mean-Field DDPG (MF-DDPG). En primer lugar, el algoritmo UCM agrupa a los UE en función de sus CSI. Luego, la asignación de potencia y descarga de tareas son resueltos por el algoritmo MF-DDPG. Los resultados demuestran una reducción del consumo de energía y del retardo de las tareas del sistema comparado con el Acceso Múltiple Ortogonal (OMA) y DQN. Además, esta propuesta mejora la velocidad de convergencia con respecto modelo DQN.

La maximización de SE y EE se atiende en [44] y [75]. La SE se refiere a la tasa de información que se puede transmitir a través de un ancho de banda dado. Los autores en [44] implementan la técnica DuQN en la MBS para la asignación de RB. El algoritmo DuQN modifica la arquitectura DQN, separa la capa final en dos flujos para estimar qué tan bueno

es para un agente estar en el estado actual y para calcular la ventaja de seleccionar las diferentes acciones en ese estado [96]. Los resultados muestran que DuQN logra un mayor rendimiento de la EE y la SE, además, se logra una convergencia de entrenamiento más rápida con respecto a los algoritmos QL y DQN. Los autores en [75] implementan el modelo DDQN para la asignación de RB en la red. A diferencia del modelo DQN, que utiliza los valores-Q máximos para seleccionar y evaluar la acción, DDQN se introduce en [97], desacoplando la selección y la evaluación en dos funciones de valor-Q. Este desacoplamiento evita la sobreestimación de los valores resultantes del uso de los valores máximos de la función de valor-Q. Una vez que el modelo está entrenado, se implementa un algoritmo de poda para reducir el tamaño de la DNN. El algoritmo de poda elimina las conexiones redundantes entre las capas totalmente conectadas. Además, reduce la complejidad del modelo DDQN, al disminuir el tiempo de inferencia, sin demeritar el rendimiento del modelo DDQN.

Los autores en [67] implementan dos esquemas de entrenamiento del modelo DDPG para minimizar la EC satisfaciendo los requisitos de latencia de las tareas de la red por medio de la descarga de cómputo y la asignación de recursos (asignación de canales, control de potencia de enlace ascendente y RA de cómputo para cada UE servido por las SBS). Ambos esquemas consideran el modelo DDPG con múltiples agentes. El esquema de aprendizaje federado implica que los agentes entrenen su modelo individualmente con su información, sin intercambiarla con otros agentes. Luego, se obtiene un nuevo modelo a partir de los parámetros de todos los agentes. Finalmente, el nuevo modelo se envía de vuelta a todos los agentes, evitando enviar información local a otros agentes en la red. Por otro lado, en el esquema de entrenamiento centralizado, los agentes envían sus experiencias (es decir, información local) a un controlador centralizado. El controlador central entrena el modelo con las experiencias de todos los agentes y, una vez entrenado, se envía de vuelta a todos los agentes para que tomen decisiones con un modelo único. Los resultados muestran un mayor consumo de energía y una garantía en los requisitos de latencia que los esquemas de aprendizaje independiente (modelos que no comparten información).

El problema de maximización de EE se aborda en [13], [14], [17], [71], [73], y [74]. En [13] se implementa un modelo DDPG para el control de potencia en una UDN de recolección de energía. En DDPG se implementan dos DNN denominadas la red-crítico y la

red-actor, respectivamente. La red-crítico aproxima la función acción-valor actualizando sus pesos mediante el error de Diferencia Temporal (TD). La red-actor actualiza la política de asignación de potencia actualizando sus pesos mediante un método de muestreo de gradiente de políticas (e.g., gradiente descendiente estocástico). Además, se implementa una red objetivo (similar a DDQN) en las redes de críticos y actores para garantizar la estabilidad. A diferencia del modelo DQN, el modelo DDPG considera un espacio de acción continuo (es decir, tasas de control de potencia continua), permitiéndole una mayor exploración. En consecuencia, este mayor espacio de acción resulta en una alta inestabilidad al inicio del entrenamiento, pero se obtiene una mayor EE cuando converge, en comparación con QL y DQN. En [14], se implementa un DQN para controlar la potencia de transmisión del enlace ascendente y maximizar la EE. En este trabajo los UE son considerados como los agentes, mientras que la función de recompensa considera la EE de todos los UE que comparten el mismo RB. Los resultados muestran un menor tiempo de convergencia y un mayor rendimiento que el modelo QL. Estos resultados se muestran de manera consistente al considerar diferentes densidades de UE y BS. Este problema de control de potencia del enlace ascendente también se atendió para una red UDN NOMA [17]. Para reducir la interferencia, se implementó un método de emparejamiento de UE con base a la diferencia de la ganancia del canal utilizado. Además, se implementó un modelo DQN para el control de potencia. Los resultados muestran que el modelo DQN propuesto supera al modelo QL con respecto al tiempo de convergencia, para diferentes densidades de UE y BS. Por otra parte, los autores en [71] implementan el modelo DQN para la UEA y asignación de potencia. Al comienzo del entrenamiento implementan un algoritmo de llenado de agua para la asignación de potencia inicial con el fin de evitar caídas de rendimiento debido a una asignación aleatoria. Los resultados muestran una mejora al utilizar está inicialización. Además, el modelo DQN logra un mayor rendimiento que el modelo QL después de unas pocas iteraciones. Este rendimiento se mantiene consistente en términos de EE a medida que aumenta la densidad del sistema, mientras que el rendimiento del modelo QL disminuye, incrementando la brecha de rendimiento entre los modelos DQN y QL. Los autores en [73] atienden el problema de RA en UDN asistido por UAV, donde el UAV actúa como una BS auxiliar. Para resolver el problema de selección del enlace de los UE y asignación de potencia se propone el modelo DQN con el objetivo de maximizar la EE. Los resultados muestran que el modelo DQN

supera los modelos QL y algoritmos heurísticos respecto a la EE, capacidad y consumo de energía. En [74], los autores proponen un modelo DDQN para el control de potencia de enlace ascendente para maximizar la EE. El sistema consiste en una red ultra-densa de redes pequeñas. El modelo DDQN es entrenado bajo el esquema de entrenamiento centralizado y ejecución distribuida. Además, se implementa un algoritmo de identificación de interferencias para modelar la información del UE y utilizarla en el estado del modelo. Los resultados muestran mayor EE y menor complejidad que los algoritmos de Programación Fraccionada (FP – fractional programming) [7] y Optimización Pseudo Convexa Sucesiva (SPCO – sucessive pseudo convex optimization) [98]. Además, los autores en [15] implementan una estrategia para controlar la interferencia en una red de celdas pequeñas ultra-densa. El modelo DQN asigna la potencia de transmisión del UE considerando la SINR de cada UE y la densidad de UE en la red. La inicialización del agente utiliza la información del agente experimentado, similar a [55], para acelerar la etapa de aprendizaje. Además, se implementa una CNN para estimar los valores-Q y comprimir el espacio de estado del agente. Este procedimiento mejora el consumo de energía y aumenta la capacidad en comparación con RL y algoritmos basados en datos.

La maximización conjunta de SE, EE y equidad se aborda en [16]. Los autores desacoplan el problema de RA en dos partes para resolver el problema con múltiples objetivos. En primer lugar, se construye la DNN para maximizar la SE. La EE y la equidad se consideran en la función de recompensa en el modelo DQN. El algoritmo propuesto decide sobre cómo asignar la potencia y los RB con información limitada de las condiciones del canal (CSI - channel state information). El modelo DQN se inicializó con valores aleatorios de sus parámetros e información parcial del CSI, lo que resultó en menos iteraciones para lograr la convergencia en comparación cuando la inicialización del modelo considera el conocimiento completo del CSI. La conclusión es que con esta propuesta solo es necesario utilizar información parcial del CSI para lograr mayor rendimiento en comparación con otros algoritmos que requieren un conocimiento completo de CSI.

Por otro lado, en [11] se implementa un modelo DQN para maximizar la SE. Los estados de cada agente consisten en su propia información, información de los vecinos interferentes y la información de los vecinos interferidos. Además, los parámetros del modelo DQN se entrenan de forma centralizada, recopilando experiencias de diferentes agentes. Es

decir, todos los transmisores de la red envían sus experiencias a un controlador central, y estos se utilizan durante el entrenamiento del modelo DQN. Una vez finalizado el entrenamiento, los parámetros del modelo DQN se actualizan y se envían a cada agente. El procedimiento anterior, reduce la memoria y los recursos computacionales de cada agente. El modelo DQN propuesto asigna los niveles de potencia en menor tiempo que los algoritmos centralizados, como el algoritmo de Error Cuadrático Medio Ponderado Mínimo (WMMSE – weighted minimum mean square error) [8] y el algoritmo iterativo basado en FP, ya que esto últimos requieren mediciones precisas y en todo momento de las condiciones del canal.

El trabajo en [66] aborda la maximización de capacidad y EE. Los autores implementan un modelo DDPG para resolver el problema de UEA y asignación de potencia. Además, implementan una capa adicional a la salida del modelo DDPG para pre-procesar la salida del modelo DDPG en variables discretas para la UEA y continuas para la asignación de potencia. Esta propuesta se evaluó en una red heterogénea inalámbrica bajo los esquemas OMA y NOMA, mostrando un mayor rendimiento al utilizar la capa de procesamiento respecto a utilizar el modelo DDPG solo para la UEA o solo para la asignación de potencia.

El problema de maximizar la capacidad de la red UDN se presenta en [18], [64], [68-70], [72] y [81-84]. En [18], se implementan dos modelos DQN para controlar la potencia de las FBS y la MBS respectivamente, y maximizar la capacidad del UE. El control de potencia se define en función de la distancia entre las FBS y la MBS. Ambos modelos DQN consideran la QoS de los FUE y MUE. Los resultados muestran que utilizar diferentes modelos DQN, uno para las FBS y otro para la MBS se logra mayor capacidad en la red en comparación si se utiliza el mismo modelo DQN en ambos agentes para controlar la potencia de los usuarios. En [64], los autores proponen el modelo BBQN para mejorar el proceso de exploración del modelo DQN. Esta propuesta se compara con las estrategias e-greedy y DQN ruidoso (noisy-DQN), consideradas como estrategias de exploración de los modelos RL. El agente que ejecuta el modelo BBQN solo considera su propio CSI. El modelo BBQN requiere de una NN bayesiana (BNN – bayesian neural network) para obtener una distribución sobre la función-valor del espacio de acción. Es decir, se aumenta la probabilidad de que el agente ejecute las acciones con mayor incertidumbre en su función-Q. A diferencia de las estrategias e-greedy y DQN ruidoso, lo anterior previene que se ejecute una exploración al azar. Los autores en [68] proponen un modelo DRL centralizado. El sistema está formado por una alta

densidad de enlaces de comunicación que conectan dispositivos directamente con otros dispositivos (pares de D2D) y por Usuarios Móviles Celulares (CMU – celular mobile users). Específicamente, implementan un modelo de optimización basado en políticas proximales (PPO – proximal policy optimization) [99] para obtener una convergencia más rápida en el modelo DDPG. PPO garantiza baja variación en el entrenamiento de la política del modelo mediante la comparación de las políticas actuales y pasadas. Los resultados se obtienen con una convergencia más rápida y de mayor capacidad que los obtenidos por métodos DRL centralizados convencionales (DDPG sin PPO). En [69], utilizan dos esquemas DDPG (centralizado y distribuido) para controlar la potencia de los UE a partir de la recolección de energía. El modelo DDPG centralizado logra resultados de asignación de potencia que resultan en mayor capacidad que el DDPG distribuido. Sin embargo, la complejidad de los enfoques centralizados aumenta exponencialmente con la densidad de las SBS. Por su parte, el modelo DDPG distribuido toma las decisiones con base a su propia información reduciendo las transmisiones de señalización requeridas para actualizar sus parámetros. Ambos esquemas DDPG obtienen mejores resultados de capacidad en la red que los enfoques DQN o codiciosos. En [70], se resuelve el problema multiobjetivo de UEA y RA mediante la implementación de un modelo DQN para maximizar capacidad de la red. Los UE son considerados como agentes, y son capaces de elegir con que radio base asociarse de entre múltiples posibles limitados a usar un RB por BS. Además, la BS divide la potencia de transmisión en partes iguales entre sus UE adjuntos. Los resultados muestran que la capacidad de la red se beneficia de más conexiones de BS y más uso de RB con respecto al método de potencia recibida de la señal de referencia máxima. Los autores en [72] atienden el problema de maximizar la QoS para una red integrada de acceso y de backhaul. Proponen un modelo DQN para resolver el control de potencia considerando las estrategias de aprendizaje independiente (IL) y cooperativo (CL). CL se utiliza para mejorar el proceso de aprendizaje a partir de las experiencias de los nodos vecinos. Los resultados de congestión, tasa de bits promedio y grado de satisfacción para diferentes densidades de UE son mejores en comparación con un modelo DQN simple.

Los autores en [81] proponen un método de asignación de niveles de potencia basado en un modelo DDPG. El modelo DDPG utiliza una CNN como función de aproximación para maximizar la capacidad de la red. Esta propuesta da como resultado una convergencia

del 39 % más rápida y una ganancia de rendimiento 14.6 % superior con respecto a las obtenidas por los modelos DPPG con DNN, DQN con DNN y DQN con CNN. Además, el modelo DDPG con CNN logra 200 veces menos tiempo computacional que el algoritmo WMMSE a costa de una capacidad ligeramente menor en escenarios altamente densos. El trabajo en [82] atiende el problema de la asignación de recursos y el cambio dinámico de banda de frecuencias en una red satelital de órbita baja ultra-densa. En este trabajo se implementó un modelo DRL para maximizar la tasa de datos del UE. Los autores implementaron un modelo DDPG para la asignación de canales y potencia considerando la ubicación del UE, las ubicaciones de los satélites y las condiciones de lluvia. Además, utilizan un algoritmo jerárquico para determinar el cambio de banda de frecuencia, lo que da como resultado un mayor rendimiento que implementar el modelo DDPG y métodos de asignación de banda aleatoria. Los autores en [83] proponen un algoritmo DDPG para maximizar la capacidad considerando la neutralidad de carbono. La neutralidad del carbono se refiere a obtener energía de alimentación autónoma para compensar el consumo de energía de las SBS. El sistema de red consiste de SBS con una fuente de energía renovable para la generación de su energía. El algoritmo propuesto supera al modelo QL con un aumento de hasta el 63% en el valor de recompensa. El trabajo futuro implica investigar la brecha de optimización del método propuesto utilizando solucionadores de optimización comerciales. En [84], los autores maximizan la capacidad de la red mediante la asignación de recursos a diferentes segmentos de red. La segmentación de la red es un modelo arquitectónico que divide una red en varios segmentos o subredes, cada uno de los cuales funciona como una pequeña red propia con la finalidad de controlar el flujo de tráfico entre las distintas subredes, en este trabajo la segmentación se realiza mediante una técnica basada en el modelo DQN. En cada segmento de red se consideran diferentes niveles de QoS cada uno enfocado en servicios mejorados de banda ancha móvil (eMBB – enhanced mobile broadband), comunicaciones ultra confiables de baja latencia (URLLC – ultra reliable low latency communications) y comunicaciones masivas tipo máquina (mMTC – massive machine-type communications). Para acelerar el aprendizaje del modelo DQN, los autores implementan una técnica de eliminación de acciones para desechar las acciones que conducen a una decisión que no cumple con los requisitos de calidad del servicio, lo que resulta en un entrenamiento de la mejor política del modelo. En cada intervalo, la acción del espacio se

filtra por URLLC seguida de un filtro por eMBB. Luego, el modelo DQN selecciona la acción en función del espacio de acción filtrado, obtenido del proceso de eliminación de acciones. Los resultados muestran una mejora de hasta un 15 % y un 10 % con respecto a un método de asignación basado en árboles de regresión y modelos basados en DQN estándar, respectivamente.

Maximizar la satisfacción del UE se refiere a satisfacer sus requisitos de ancho de banda [39] y QoS [40]. En [39], la UEA y la asignación de RB se realizan mediante un modelo ACDL con Recolección de Energía (EH – energy harvesting). El modelo ACDL, es similar al modelo DDPG, consiste de dos redes; red-actor y red-crítico. Después de cada acción, la BS informa el número de UE satisfechos y el nivel de energía de la batería del UE para actualizar los estados del entorno en el controlador central. Luego, las dos redes actualizan sus funciones a partir del error de Diferencia Temporal (TD – temporal difference). Los resultados demuestran que se logra una menor convergencia comparado con otros enfoques de aprendizaje. Sin embargo, dado que la capacidad de la batería es finita y el ancho de banda está restringido, los resultados muestran que a medida que aumenta el número de UE, la capacidad de la red decae y se reducen las diferencias en los resultados de capacidad obtenidos entre los diferentes enfoques evaluados (convergen a resultados similares). Por otro lado, el modelo DQN se implementa para asignar los RB en redes UDN en el trabajo [40]. Los autores implementaron un barrido priorizado [100] y un mecanismo heurístico en la arquitectura DQN para acelerar la convergencia del algoritmo. El esquema de barrido priorizado asigna prioridad a las experiencias generadas por la interacción entre el agente y el entorno, con el fin de muestrear los estados del buffer de ER que tienen una mayor probabilidad de cambiar los valores-Q de la red. El mecanismo heurístico ayuda a establecer si la acción generada se basa en un algoritmo de aproximación tradicional, esto se utiliza para mejorar la estrategia de selección de la acción. Los resultados muestran que el mecanismo heurístico ofrece mejor rendimiento (en condiciones de tráfico ligero y tráfico pesado) que los algoritmos de programación tradicionales como round-and-robin, equidad proporcional y max C/I.

En [43], los autores afirman que es un desafío obtener una estrategia óptima para lograr conjuntamente la Asociación de Usuarios a una nueva BS (UEA) y la asignación de recursos. Los autores maximizan la utilidad de la red utilizando un modelo D3QN. La función

de utilidad es definida como diferencia entre la ganancia y el costo de transmisión respecto a la capacidad de la red y a la potencia de transmisión. La propuesta consiste en un modelo DDQN y una arquitectura de duelo (DuQN) para abordar la estimación del valor-Q demasiado optimista y para una mejor evaluación de políticas, respectivamente. Además, para reducir la complejidad del amplio espacio de acción, se implementa un DRL de múltiples agentes cooperando con mensajes de señalización entre los UE para recopilar la información global del estado y recopilar las políticas de todos los UE. Los resultados indican que el rendimiento del modelo D3QN supera al modelo DQN en escenarios con altas densidades de UE. Además, los resultados muestran que otros algoritmos, como el algoritmo genético y el de potencia máxima de la señal de recepción, no logran encontrar estrategias adecuadas para cumplir con los requisitos de los UE cuando el número de UE y los niveles de QoS aumentan, comparados con los enfoques DRL.

2.3.4 Problemas de investigación abordados por modelos basados en aprendizaje automático

La Figura 2.5 muestra la forma en la que están distribuidos los trabajos seleccionados para esta revisión según el problema de RA que se abordó. De los trabajos analizados en esta revisión, controlar la interferencia mediante el control de potencia en la red UDN es el problema de investigación mayormente estudiado [11] , [13], [15], [17], [18], [49], [51], [53-55], [57], [60], [61], [64], [68], [69], [72], [74], [77-81], [83], seguido de la asignación de RB [40], [41], [44-46], [48], [52], [58], [63], [75], [84]. Otros trabajos proponen modelos de RA más complejos en los que incluyen dos o más objetivos de optimización como la UEA y asignación de potencia conjunto [14], [42], [56], [66], [71], [73], [76], la asignación de potencia y RB conjunto [16], [50], [62], [67], [82], la UEA y asignación de RB conjunto [39], [43], [70], el agrupamiento de celdas y asignación de potencia conjunto [59], [65], el agrupamiento de celdas y asignación de RB conjunto [47], y el agrupamiento de celdas, asignación de potencia y RB conjuntos [85].

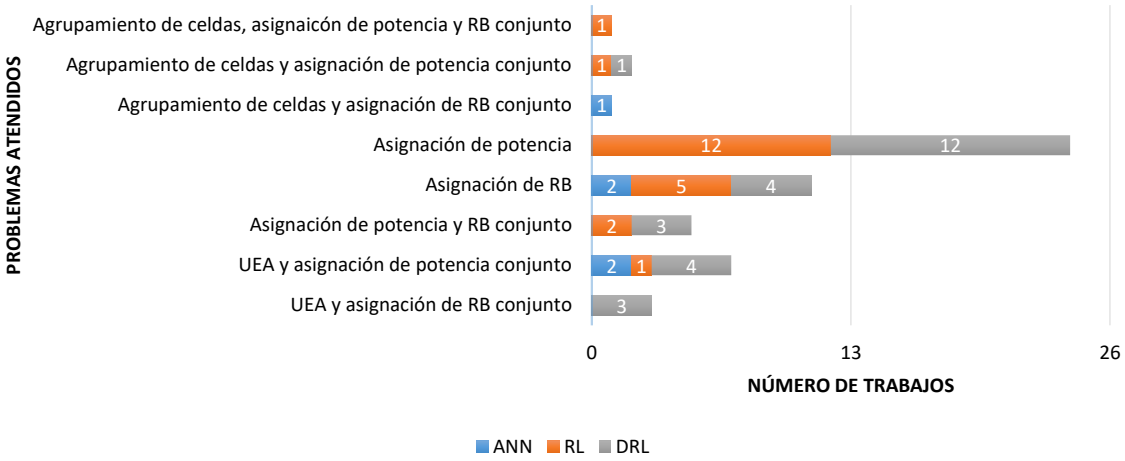

Figura 2.5 Distribución de los trabajos seleccionados según el problema atendido para la asignación de recursos en redes ultra-densas.

2.3.5 Métricas clave de rendimiento en modelos basados en aprendizaje automático

En total se identificaron doce indicadores de rendimiento (KPI) que los trabajos analizados utilizan como referencia para establecer la estrategia de cómo asignar los recursos en la red UDN. En la Figura 2.6 se pueden observar estos indicadores de rendimiento que van desde el retardo, consumo de energía, eficiencia energética, equidad, interferencia, tasa de interrupción, tasa de pérdida de paquetes, SINR, SE, capacidad, satisfacción de UE hasta la utilidad de la red. También se muestran el porcentaje de los trabajos analizados que toman en cuenta estos KPIs, Por ejemplo, se observa que el 54% de los documentos analizados se enfocan en mejorar la capacidad (37%) o la eficiencia energética (17%), mientras que la interferencia, tasa de interrupción (outage) y la Relación Señal-Interferencia (SINR) son los menos utilizados como indicador de rendimiento. Lo anterior tiene sentido ya que si estos indicadores se controlan de alguna manera tendrán un efecto sobre la capacidad o eficiencia energética de la red o de sus usuarios. Además, la Tabla 2.8 muestra una lista de los trabajos y el KPI que analizan según el grupo ML al que pertenecen.

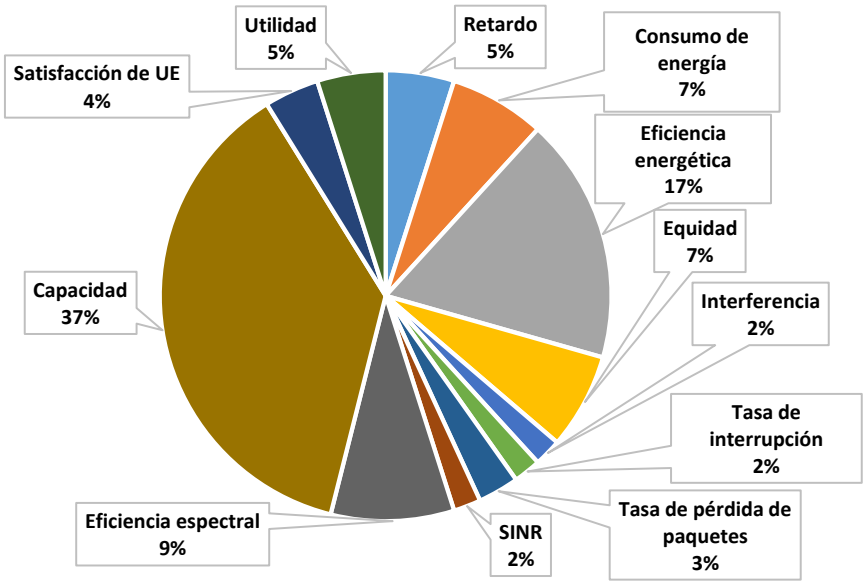

Figura 2.6. Distribución de los indicadores de rendimiento clave (KPI) considerados para la asignación de recursos en redes ultra-densas.

2.4 Discusión

Dada la naturaleza NP-hard del problema de asignación de recursos, encontrar la mejor solución, es decir, aquella que asigne, a los nodos y usuarios móviles, los canales y el nivel de potencia para que la red móvil logre su óptimo KPI, requiere, primero, enumerar y, después, evaluar todas las soluciones posibles del problema, lo cual es impráctico incluso para redes con baja densificación de BS. Las técnicas de Aprendizaje Automático (ML) tienen el potencial de resolver de manera eficiente problemas no estructurados y aparentemente intratables. En este capítulo se ha mostrado evidencia de que los modelos de ML son capaces de encontrar soluciones con la mejor asignación de recursos y óptimo KPI en escenarios de red ultra-densas (UDN) en comparación con los algoritmos exactos y de aproximación. Sin embargo, la capacidad de cómputo limitada por las entidades que ejecutan los algoritmos de RA no permite implementar algoritmos de gestión de recursos aún más complejos. En este sentido, los métodos de poda, tal como se implementan en [75], reducen el tamaño del modelo de los modelos basados en ANN y, por lo tanto, su complejidad. Sin embargo, se requiere más trabajo de investigación sobre estos modelos de ML evaluados en redes de alta densidad con el fin de establecer un equilibrio entre la complejidad del modelo, la capacidad del sistema y el tiempo ejecución.

Tabla 2.8. Clasificación de los trabajos analizados por modelo de aprendizaje automático ML y por indicador de rendimiento clave (KPI).

Enfoque de ML	Referencia	Indicadores de rendimiento claves (KPI)											
		Retardo	Consumo de energía	Eficiencia energética	Equidad	Interferencia	Tasa de interrupción	Tasa de pérdida de paquetes	SINR	Eficiencia espectral	Capacidad	Satisfacción de UE	Utilidad
DRL	11												✓
	13, 14, 18, 71			✓									
	15		✓			✓					✓		✓
	16											✓	
	17			✓	✓					✓			
	39									✓	✓		
	40, 66, 74			✓							✓		
	43, 82										✓		✓
	44, 75			✓						✓			
	64, 68, 69, 70, 81, 84										✓		
	65, 67	✓	✓										
	72										✓	✓	
	73		✓	✓							✓		
	83		✓								✓		
ANN	41, 45							✓			✓	✓	
	42			✓									
	47									✓			
	76										✓		
RL	46								✓		✓		
	48, 63	✓			✓						✓		
	49, 51, 53, 58, 60, 61, 78										✓		
	50, 56			✓									
	52	✓									✓		
	54			✓				✓		✓			
	63		✓			✓					✓		✓
	65		✓							✓	✓		
	67				✓						✓		
	70				✓		✓		✓	✓	✓		
	85									✓	✓		
	87		✓		✓						✓		
	88			✓	✓		✓						
	93			✓							✓	✓	

Por otro lado, implementar modelos de ML más sofisticados para atender las necesidades de las redes UDN también es un desafío, debido a que estas técnicas requieren entrenamientos que les toman tiempos más prolongados. A pesar de que estos modelos ML robustos son entrenados mediante simulación antes de su implementación, las futuras redes inalámbricas presentarán comportamientos altamente impredecibles (por la coexistencia de entidades heterogéneas, CMU, D2D, IoT o DID), y entornos de red dinámicos (por el tráfico cambiante, la movilidad de UE, o la conexión/desconexión del UE). Por lo que, lograr que la decisión del modelo de ML sea el adecuado para el entorno de red actual requerirá de una

recopilación de conjuntos de datos eficiente y un entrenamiento adicional constante. Para resolver lo anterior, algunos trabajos implementan estrategias de Transferencia de Conocimiento (TL - transfer learning) para gestionar el conocimiento aprendido durante diferentes condiciones de red. El conocimiento se transfiere para acelerar la convergencia de decisión del modelo, esta puede ser en forma de tablas-Q [51], [61], [62], [79], valores-Q [53], [55] o parámetros del modelo [15], [16], [76]. Sin embargo, como se mencionó anteriormente, las condiciones de cada SBS pueden ser muy diferentes entre sí, provocando que se genere información diversa de la interacción entre el agente y el entorno. Utilizar la información de otras SBS directamente sin analizarla previamente, podría provocar caídas de desempeño en lugar de acelerar el entrenamiento de los modelos. Para aprovechar de mejor manera la diversidad de la información generada, es necesario implementar métodos que utilicen esta información con base a los efectos en el aprendizaje, para acelerar el entrenamiento o añadir robustez al modelo.

Las futuras redes inalámbricas B5G esperan niveles de QoS más exigentes y velocidades de transferencia datos más altas, con dispositivos de uso intensivo de datos [48] y de retardo crítico [63], por lo que deberán ejecutar de manera eficiente las estrategias de RA para controlar su acceso a la red. Por lo tanto, se requieren metodologías novedosas para hacer frente a las demandas de servicio pronosticadas para las redes futuras. Por ejemplo, metodologías como considerar diferentes diseños de modelos para los agentes MBS y SBS (como en [11]) en lugar de implementar un diseño de modelo único para todos los agentes. Lo anterior ocasionará que cada tipo de red se beneficie de sus características de forma distribuida, con traspaso de mensajes para obtener información de estado sobre sus vecinos. En este sentido, los UE no críticos podrían centrar su función de recompensa en mantener su nivel de QoS para liberar recursos excedentes que podrán utilizar los UE críticos, como aquellos usuarios que requieren tasas de datos alta y retardo mínimo.

El problema de asignación de potencia es uno de los temas de estudio más relevantes para las redes UDN. Por un lado, para controlar la interferencia a causa de la compartición de espectro o del re-uso de bandas de frecuencia, pero por otro, las SBS consumen energía incluso en modo de suspensión (sleep mode). En este modo, las SBS transmiten información de señalización para gestionar las actividades que se presenten en la red en cualquier instante, tales como completar procesos de asociación de UE a BS, encendido/apagado de SBS o

movilidad del espectro. La densificación de la red conlleva a una mayor complejidad al administrar información por el incremento de las transmisiones de señalización y, a medida que aumenta el número de SBS, aumenta la probabilidad de mayor interferencia en las transmisiones, provocando una degradación de la capacidad de la red.

2.5 Problemas abiertos

En esta sección se presentan los problemas abiertos identificados que requieren más investigación respecto a la implementación de aprendizaje automático para la asignación de recursos en redes ultra-densas.

2.5.1 Heterogeneidad de redes ultra-densas

La evolución de la red inalámbrica conlleva a la coexistencia de varios dispositivos, como IoT, DID o D2D. Esta coexistencia provoca un comportamiento impredecible en la red debido a que estos dispositivos requieren diferentes niveles de QoS, mientras se conectan y desconectan continuamente. Por otra parte, los objetivos de las BS difieren entre las MBS, PBS, FBS, UAV o EH-BS. Lo anterior requiere diferentes diseños de modelos de ML para cada una de estas entidades de red. A pesar de esta heterogeneidad, el conocimiento de los diferentes modelos se puede aprovechar para aprender y adaptarse eficientemente a las diferentes condiciones y requisitos de red. Como se mencionó en esta revisión, el conocimiento se explota simplemente mediante la transferencia de conjunto de datos o parámetros de red. No obstante, aún no se explora la transferencia de conocimiento entre diferentes diseños de modelos de ML (es decir, modelos de ML de MBS a UAV o de IoT a UE), por ejemplo, técnicas como la destilación de conocimiento para extraer las características intrínsecas del modelo.

2.5.2 Escalabilidad de los modelos de aprendizaje automático

Los modelos de ML deben diseñarse para ser escalables. A medida que la red se densifica, las soluciones centralizadas se vuelven inviables. Los esquemas distribuidos se vuelven más factibles frente a la densificación de la red. Sin embargo, la falta de CSI puede empeorar el rendimiento de la red debido a comportamientos egoístas al considerar poca información local. Por lo que se necesita diseñar cuidadosamente los modelos de ML con el

fin de evitar degradar el rendimiento de la red, incluyendo la cooperación o la conciencia de interferencia para reducir la interferencia potencial hacia otras entidades de red. El desafío sigue siendo determinar la información más relevante a considerar para alimentar los modelos de ML para evitar gastos generales innecesarios y evitar instancias de entrenamiento prolongadas.

2.5.3 Diseño de los modelos de aprendizaje automático

El tiempo de inferencia para la toma de decisiones depende del diseño del modelo, por ejemplo, el número de capas y neuronas en la DNN. A pesar de que modelos con más capas pueden extraer información valiosa para construir modelos robustos, las redes inalámbricas necesitan tiempos de respuesta rápidos, tal como se prevé en los servicios de URLLC. Sin embargo, cuanto más densa es la DNN, menor plasticidad tiene el modelo para adaptar sus parámetros. Por lo tanto, una compensación entre la cantidad de capas, la adaptabilidad para el aprendizaje y el tiempo de ejecución requieren más investigación para analizar la confiabilidad de los modelos de ML para diferentes tipos de servicios.

2.5.4 Diversidad del conjunto de datos

En los distintos sistemas de red se genera una gran cantidad de información. Esta información se procesa como entradas para los algoritmos de ML. Sin embargo, este proceso conduce a una pérdida de recursos debido a los altos requisitos de señalización. Por ejemplo, en lugar de obtener sistemáticamente toda la información sobre la marcha, es posible que se requiera un mejor análisis de datos para identificar la información más relevante relacionada con el aprendizaje de la toma de decisiones de los algoritmos de ML para los problemas de RA. Sin embargo, el reto del análisis de la información procesada recae en cómo medir la calidad de estos conjuntos de datos y como medir su impacto en el rendimiento de la red con el fin de aprender de manera eficiente mientras se reducen las instancias de entrenamiento y mejora el rendimiento de la red.

2.5.5 Consumo de energía

Además del consumo de energía de las BS y los UE, a menudo no se considera el impacto del consumo de energía de las estrategias de entrenamiento fuera de línea y en línea,

lo que plantea la cuestión de los efectos ambientales y las limitaciones de la batería de los dispositivos que ejecutan estos modelos de ML. Los modelos robustos requieren DNN densos para extraer las características que a su vez requieren instancias de entrenamiento más largas. Al mismo tiempo, las DNN poco profundas se adaptan más rápidamente a costa de actualizaciones de entrenamiento adicionales debido a la falta de robustez. Por lo tanto, los esquemas de ML futuros deben considerar el consumo de energía y las limitaciones de hardware dentro de su diseño de modelo para la implementación en redes B5G.

2.6 Resumen la revisión sistemática

Esta revisión estructurada de la literatura brinda información sobre las características de diseño de los algoritmos de ML para la asignación de recursos en las UDN. Se extrajeron la información de los mecanismos de diferentes modelos de ML, así como los objetivos de optimización, los escenarios de red en que fueron implementados y sus KPI implementados para evaluar el rendimiento de la red. La RA es un problema desafiante en UDN. Sin embargo, la implementación de algoritmos inteligentes ayudará a la red a encontrar soluciones confiables para la asignación de recursos con un menor tiempo de ejecución y con un mejor rendimiento que los algoritmos exactos y de aproximación. Por otra parte, la implementación de modelos de ML con diferentes objetivos en la misma red muestra una estrategia prometedora que necesita más investigación para definir el potencial de este enfoque. Por último, la tendencia en DRL podría llevar a resultados interesantes en redes futuras para abordar el desafío de alta dimensionalidad de las UDN.

Con base al análisis de este capítulo, una de las características más relevantes para el diseño de los modelos de ML es su inicialización, ya que los modelos de ML requieren adaptarse rápidamente para distintos tipos de aplicaciones. Sin embargo, para adaptarse rápidamente a los cambios de los sistemas B5G, los modelos de ML requieren de información relevante del entorno para ajustar eficientemente sus parámetros ante las condiciones de red que se presenten. Dadas las dinámicas de los sistemas B5G, los modelos DRL se muestran como los modelos más adecuados para adaptarse a los cambios que se presenten en el entorno de manera rápida y con un mayor rendimiento que los modelos de RL. Con base a los problemas abiertos identificados, este trabajo de tesis aborda los siguientes problemas abiertos: 1) Diseño de los modelos de aprendizaje automático: a medida que las entidades en

la red incrementan, el problema de asignación de recursos se vuelven más complejos, mientras que, los sistemas B5G requieren modelos con menor tiempo de respuesta y mayor adaptabilidad ante las fluctuaciones de la red; y 2) Diversidad del conjunto de datos: las experiencias que se generen en los sistemas B5G dependerán de las condiciones dinámicas del entorno, provocando que se almacenen ciertas experiencias poco relevantes para el aprendizaje, prolongando el entrenamiento de los modelos. Por lo anterior, en este trabajo de tesis se considera un diseño del modelo DQN (presentado en el Capítulo 4) donde el tiempo de respuesta del modelo no se ve afectado a medida que se modifica el número de usuarios o BS en la red. Además, se aprovecha el conocimiento (parámetros y experiencias) de otros modelos entrenados en entornos de red similares para inicializar el modelo DQN por medio de un método de transferencia de aprendizaje. Sin embargo, esta transferencia de conocimiento puede degradar el aprendizaje del modelo DQN si los parámetros o experiencias transferidas no son adecuados a las nuevas condiciones de red. Por lo que, como se mencionó en el capítulo 1, para acelerar el aprendizaje del modelo DQN e incrementar el rendimiento de la red B5G, se propone un método de gestión de experiencias del modelo DQN con un buffer dual para diversificar las experiencias recolectadas y reutilizar las experiencias relevantes durante la fase de entrenamiento del modelo.

Capítulo 3

Aprendizaje por refuerzo profundo

La asignación de recursos en entornos inalámbricos y móviles que varían con el tiempo se vuelven problemas complejos de resolver debido a que la manera en la que se asignen los recursos debe coincidir con las condiciones del canal para eficientizar el uso de los recursos radioeléctricos. Realizar esta tarea de asignación de recursos se dificulta cuando el número de UE y SBS incrementan como se espera que suceda en redes B5G.

En este capítulo se presenta el funcionamiento general de los modelos de aprendizaje por refuerzo profundo. Estos modelos aprenden políticas de asignación de recursos que se adaptan a las condiciones variantes de las redes móviles B5G. Después, se introduce el modelo Deep Q-Network (DQN) y los elementos que lo componen. El modelo DQN es ejecutado por la entidad de red que controla los recursos radioeléctricos y es el encargado de tomar las decisiones de asignación de recursos. Para que el modelo DQN aprenda la mejor política debe entrenarse para que autoajuste sus parámetros de operación. En la última parte del capítulo se introduce la estrategia de transferencia de conocimiento que acelera la etapa de entrenamiento de los modelos DRL, la cual es utilizada en los métodos de este trabajo de investigación descritos en el Capítulo 4.

3.1 Proceso de decisión de Markov

La asignación dinámica de recursos se define como un problema de decisión secuencial en el que la entidad encargada de asignar los recursos lo hace a partir de observar

la información del entorno. En DRL, el entorno de aprendizaje comprende todo aquello con lo que interactúa el agente. Por ejemplo, en una red móvil celular que consiste de una MBS y de varias SBS, si la MBS es el agente, el resto de las SBS son consideradas como el entorno de aprendizaje. En este trabajo se utiliza indistintamente el término de entorno y red celular para referirse al entorno de aprendizaje.

Para resolver este problema de decisión secuencial por medio de los modelos de DRL, primero se formula el problema como un Proceso de Decisión de Markov (MDP – markov decision process). El proceso de MDP describe la interacción entre el agente y el entorno [101], como se muestra en la Figura 3.1. El MDP se define como $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ en donde $\mathcal{S}$ representa el conjunto de todos los estados del entorno, $\mathcal{A}$ representa el conjunto de todas las acciones disponibles por el agente y $\mathcal{R}$ representa la recompensa inmediata obtenida de tomar la acción $a \in \mathcal{A}$ en el estado $s \in \mathcal{S}$. Las decisiones (acción) se toman a partir de la política del modelo. La política del modelo es una función que regresa un valor para cada acción posible. La acción de valor más alto corresponde a la mejor acción. Sin embargo, está política requiere adaptarse a las condiciones del entorno. Adaptar está política requiere de un proceso de entrenamiento que consiste en un proceso iterativo de interacciones entre el agente y el entorno, en el que el agente toma diferentes acciones en cada estado visitado con el fin de aprender la mejor política π. Durante el entrenamiento, se evalúa el valor de recompensa hasta que se alcanza la convergencia en el valor de recompensa que indica que el modelo aprendió la política óptima π^*, es decir, seleccionar la mejor acción posible del entorno en que fue evaluada.

El proceso de entrenamiento se describe de la siguiente forma. En un instante t, el agente recolecta la información del entorno $s^t \in \mathcal{S}$. Con base en el estado actual, el agente toma una acción $a^t \in \mathcal{A}$ siguiendo la estrategia de exploración, i.e., seleccionar la mejor acción conocida o seleccionar una acción aleatoria dentro del espacio de acción. Esta acción induce una transición en el entorno generando un nuevo estado $s^{t+1} \in \mathcal{S}$ y un valor de recompensa r^t. Al final de la interacción se genera una experiencia $e^t = (s^t, a^t, s^{t+1}, r^t)$ que define la transición del entorno. Estas experiencias se almacenan en un buffer de ER $\mathcal{D}$, las cuales se utilizan durante el proceso de aprendizaje o entrenamiento descritos en las siguientes secciones.

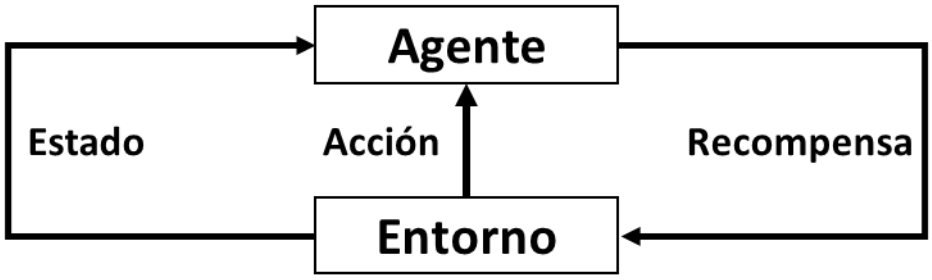

Figura 3.1. Interacción de los modelos de aprendizaje por refuerzo entre el agente y el entorno.

3.2 Deep Q-Network

En este trabajo se implementó el modelo Deep Q-Network (DQN) [20] que es uno de los algoritmos más utilizados entre los algoritmos de DRL. Para calcular el valor de recompensa durante el entrenamiento, el modelo DQN utiliza la recompensa cumulativa a largo plazo:

$$R^t = \sum_{t=0}^{\infty} \zeta r^t \tag{1}$$

donde ζ es un parámetro $0 \leq \zeta \leq 1$ utilizado para determinar el valor presente de las recompensas futuras, llamado factor de descuento [102]. Para evaluar la política óptima π^* del modelo DQN se utiliza la función acción-valor:

$$Q(s^t, a^t; \theta^t) = \mathbb{E}[R^t | s^t = s, a^t = a], \tag{2}$$

donde $\mathbb{E}[\bullet]$ es el operador del valor esperado y θ^t son los parámetros de la DNN en el tiempo t. La DNN es utilizada por los modelos DQN para aproximar la función acción-valor. La función acción-valor devuelve el valor de recompensa esperado en función del estado actual y de la acción tomada en el tiempo t. Como se mencionó anteriormente, la política del modelo requiere adaptarse debido a que inicialmente la función acción-valor devuelve valores arbitrarios que dependen de la inicialización de los parámetros θ^t. Parar lograr que los valores de la función acción-valor sean más exactos y se aproximen a los valores óptimos, i.e., $Q^*(s^t, a^t; \theta^t)$, se realiza una actualización de los parámetros θ^t cada T intervalos de entrenamiento utilizando una función de pérdidas. La función de pérdidas mide la diferencia entre el valor de la política actual y el valor de referencia esperado. Por ejemplo, la función de pérdidas a partir de una experiencia $e^t = (s^t, a^t, s^{t+1}, r^t)$ es calculada como:

$$\mathcal{L}(\theta^t) = \left(y^t - Q(s^t, a^t; \theta^t)\right)^2, \tag{3}$$

en donde y^t es el valor máximo de las recompensas futuras (valor de referencia esperado) en tiempo t, calculado con la ecuación de Bellman [20], definida como:

$$y^t = r^t + \zeta \max_{a'}(s^{t+1}, a'; \theta^t) \tag{4}$$

Por último, los parámetros de la DNN se optimizan utilizando un método de gradiente descendente [103] para minimizar la función de pérdidas (3). Los parámetros se actualizan a partir de la ecuación (5):

$$\theta^{t+1} = \theta^t + \eta\left(y^t - Q(s^t, a^t; \theta^t)\right)\nabla Q(s^t, a^t; \theta^t) \tag{5}$$

donde η es la tasa de aprendizaje que nos permite ajustar la magnitud del paso en cada actualización.

3.2.1 Red Neuronal Profunda

Los modelos DQN utilizan las DNN para aproximar la función acción-valor. Los componentes principales de la DNN se muestran en la Figura 3.2. La DNN consiste en la capa de entrada, capas ocultas y capa de salida. La capa de entrada recibe la información del estado (definida en la Sección 4.2). Los valores de entrada s se propagan hacia adelante para calcular los valores $Q(s^t, a^t; \theta^t)$ en la capa de salida. Por ejemplo, el valor de salida de una neurona se calcula con la información de salida de la capa anterior. Suponiendo que la capa anterior tiene M_l neuronas, la información y los parámetros recibidos de la capa anterior están definidos por los vectores $\boldsymbol{v} = \{v_1, \dots, v_{M_l}\}$ y $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_{M_l}\}$, respectivamente. Entonces, el valor de salida de cada neurona se calcula de la siguiente forma:

$$salida = \varphi\left(b + \sum_{m=1}^{M_l} v_m \theta_m\right) \tag{6}$$

donde b es el valor de sesgo y φ es una función de activación utilizada para aproximar funciones complejas (no lineales) en las DNN. A partir de (6), cada neurona obtiene los valores de la función acción-valor para estimar la mejor acción en el modelo DQN.

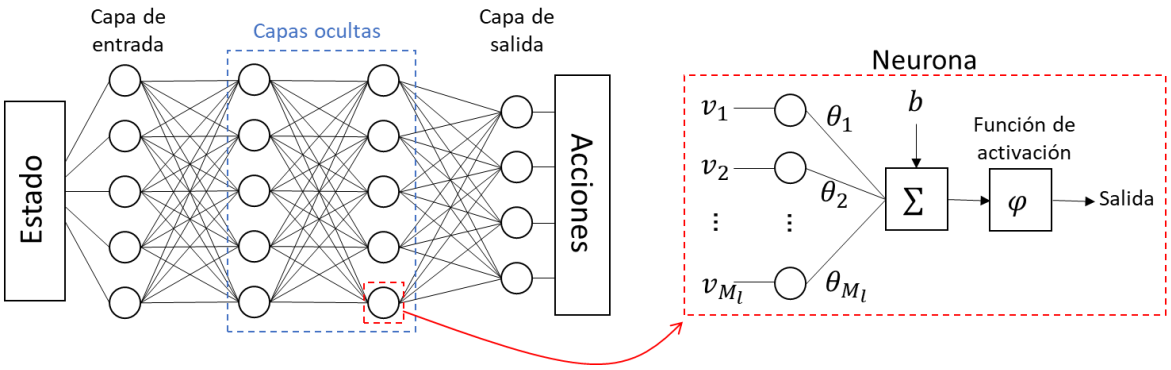

Figura 3.2. Ejemplo red neuronal profunda.

3.2.2 Red Neuronal Profunda objetivo

Uno de las limitantes del modelo DQN es la inestabilidad causada por calcular los valores de la función acción-valor actual y los valores máximos de recompensa futuros (siguiendo la ecuación de Bellman) con la misma DNN [20]. Lo anterior, ocasiona que el error de la función de pérdidas (3) cambie significativamente en cada intervalo de entrenamiento. Una solución a lo anterior es utilizar una DNN adicional llamada DNN objetivo [97]. Esta DNN objetivo mantiene sus parámetros θ_-^t fijos para generar los valores máximos de recompensa futuros. Al inicio del entrenamiento los parámetros de las DNN y DNN objetivo son idénticos. Los parámetros de la DNN se actualizan cada intervalo de entrenamiento, mientras que los parámetros de la DNN objetivo se mantienen fijos y se duplican periódicamente (i.e., $\theta_-^t \leftarrow \theta^t$) cada L intervalos de entrenamiento. Utilizar las dos DNN durante el entrenamiento permite contar con un entrenamiento más estable ya que las estimaciones se vuelven más precisas al tener un punto de referencia fijo con la DNN objetivo.

3.2.3 Buffer de repetición

El buffer de ER se introduce en [21] para evitar la correlación entre experiencias consecutivas. Es decir, durante el entrenamiento, el modelo aprende utilizando las experiencias del buffer de ER. Esto es, con base en cada experiencia, el modelo DQN calcula el valor de la política actual y el valor de referencia esperado (ver Sección 3.2). En caso contrario, al no utilizar el buffer de ER, el modelo corre el riesgo de dejarse influenciar por las condiciones más recientes del entorno ocasionando un sesgo en el aprendizaje. Por lo que las experiencias se almacenan en un buffer de gran capacidad y en cada intervalo de

entrenamiento se seleccionan al azar un mini-lote de experiencias que eliminan la correlación entre las experiencias utilizadas durante el ajuste de los parámetros del modelo DQN.

Convencionalmente, para gestionar las experiencias, se utiliza un buffer tipo FIFO (*First-in-First-Out*) en donde se almacenan las experiencias de manera secuencial. Una vez que el buffer de ER está lleno, las experiencias con mayor antigüedad se reemplazan por las más recientes. Mantener las experiencias en un buffer de ER incrementa la eficiencia de las experiencias obtenidas durante la interacción entre el agente y el entorno, ya que estas se reutilizan constantemente. Además, almacenar experiencias antiguas permite preservar el conocimiento de estados previos que pueden utilizarse para ajustar los parámetros del modelo. Por lo que la política del modelo DQN puede tomar decisiones adecuadas ante una gran cantidad de condiciones de estado. Considerando la DNN objetivo y los mini-lotes de experiencias, la función de pérdidas (3) es reemplazada por:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(r^t + \zeta \max_{a'}(s^{t+1}, a'; \theta_-^t) - Q(s^t, a^t; \theta^t)\right)^2\right] \tag{7}$$

3.2.4 Exploración y explotación

Durante la fase de entrenamiento del modelo DQN es necesario tomar diferentes acciones para explorar el entorno. Por ejemplo, tomar siempre la acción correspondiente al valor máximo de la función acción-valor evitará que el agente DQN visite nuevos estados que pueden llevar a mejores óptimos locales o al óptimo global. Usualmente los algoritmos DQN siguen la política de exploración épsilon-codicioso o ε-greedy, como se muestra en (14):

$$\pi(s, a) = \begin{cases} acción\ aleatoria, & \varepsilon \\ \max\limits_{a \in A} Q(s^t, a; \theta^t), & 1 - \varepsilon \end{cases} \tag{8}$$

donde el valor ε indica si el modelo tomará la mejor acción conocida hasta el momento siguiendo la política del modelo, i.e., $\pi(s, a)$ o si el modelo tomará una acción al azar con base a las acciones disponibles.

Al inicio del entrenamiento, la red DNN se inicializa al azar y es necesario ajustar la política del modelo con base en la información del entorno. Contar con un valor alto de épsilon permite recolectar información del entorno y mejorar el aprendizaje del modelo DQN. La recolección de información del entorno o generación de experiencias por medio de

acciones al azar se conoce como exploración del entorno. En la fase de exploración del entorno se generan diversas experiencias que permiten adaptar de mejor manera la política del modelo. A medida que el entrenamiento avanza, el valor de épsilon disminuye, como se muestra en la Figura 3.3. Conforme avanza el entrenamiento y se alcanza la convergencia del modelo, las acciones al azar se vuelven innecesarias si el entorno se mantiene relativamente estable. Sin embargo, si el entorno cambia constantemente, mantener un valor pequeño de épsilon (mayor exploración de experiencias) permite adaptar la política a los cambios que se presenten en el entorno.

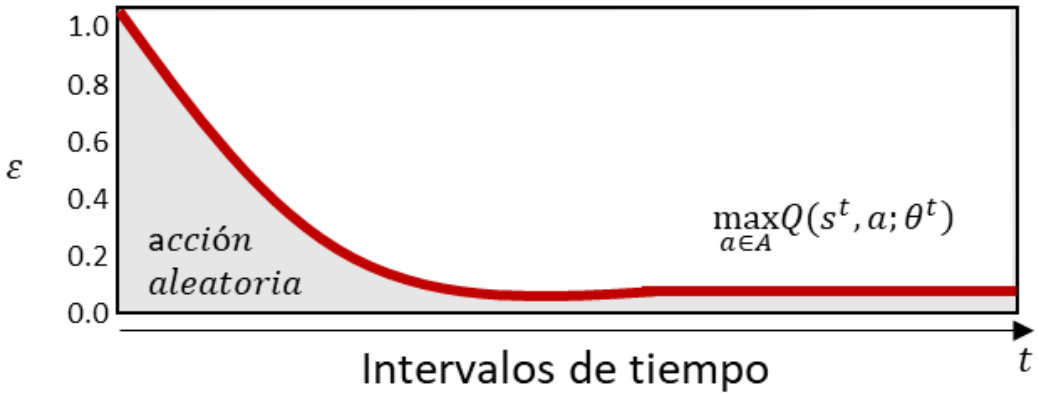

Figura 3.3. Decaimiento del valor de épsilon durante la fase de entrenamiento respecto a los intervalos de tiempo.

La interacción entre los elementos DQN descritos en esta sección para un ciclo de entrenamiento se muestra en la Figura 3.4. Para simplificar el diagrama, se consideró que el mini-lote es igual a uno. Sin embargo, en la práctica la función de pérdidas recibe la información de K experiencias (tamaño del mini-lote) para actualizar los parámetros de la DNN. El proceso inicia con el almacenamiento de experiencias en el buffer de ER. Después, se muestrea el mini-lote del buffer de ER y se calculan los valores de la función acción-valor y la función acción-valor esperada con la DNN y la DNN objetivo, respectivamente. Con los valores calculados, se actualiza la DNN con base a la función de pérdidas. Por último, cada L ciclos de entrenamiento, los parámetros de la DNN objetivo se igualan a los parámetros de la DNN.

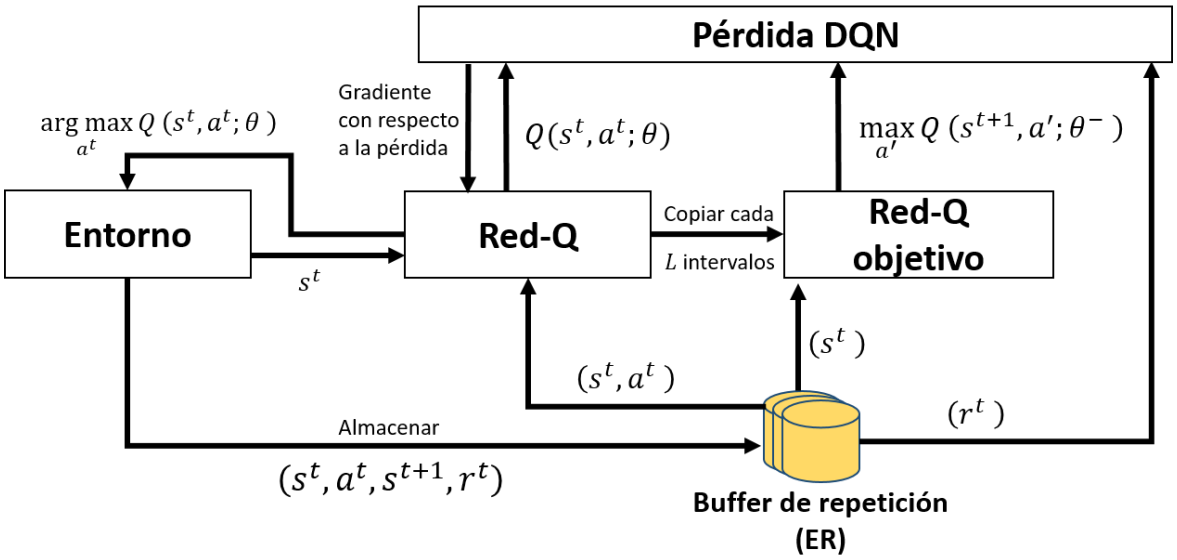

Figura 3.4. Componentes del algoritmo DQN.

3.3 Aprendizaje por Transferencia

Como se mencionó en las secciones anteriores, es necesario realizar un proceso de entrenamiento para encontrar la política óptima para cada condición del entorno de red. El proceso de entrenamiento de un modelo con parámetros inicializados al azar y sin conocimiento de antemano es conocido como entrenamiento desde cero o Scratch. Este entrenamiento se vuelve inviable para un despliegue en un entorno real debido a que compromete la seguridad y confiabilidad del sistema inalámbrico ya que las acciones al azar en la fase de exploración provocarán interferencia en el sistema y no cumplirán con los requisitos de QoS de los usuarios. Por lo anterior, primero se entrenan los modelos bajo simulación y después se transfiere su conocimiento a otros agentes en entornos similares mediante la Transferencia de Conocimiento (TL) [104]:

$$\theta_{aprendiz} = \theta_{experto} \tag{9}$$

$$\mathcal{D}_{aprendiz} = \mathcal{D}_{experto} \tag{10}$$

donde el agente aprendiz utiliza el conocimiento del agente experto. Este TL directo puede ser utilizado para añadir información de diferentes modelos DQN. Por ejemplo, modelos con diferentes distribuciones espaciales de tráfico [17] o entornos con topologías de red dinámicas como con nuevos nodos desplegados o reconfiguraciones de red [105].

Sin embargo, debido a las diferencias entre los entornos de implementación y los entornos de los agentes expertos, es necesario realizar un entrenamiento de ajuste para adaptar el modelo DQN a las nuevas condiciones del entorno de implementación. En caso de que las características de los entornos de la red móvil sean diferentes en cada instante de tiempo, utilizar TL podría ocasionar un aprendizaje negativo ya que las políticas entrenadas para otras tareas similares pueden no ser completamente adecuadas. Por lo que en los siguientes capítulos se evalúan los efectos de aprendizaje para un modelo DQN que reutiliza el conocimiento de otros modelos por medio de TL y se propone un mecanismo de gestión para acelerar el entrenamiento de ajuste y mejorar el rendimiento del modelo.

Capítulo 4

Mecanismos de gestión de experiencias para modelos Deep Q-Network

En este capítulo se presenta el esquema de entrenamiento del modelo DQN implementado para la asignación de potencia en redes celulares B5G. Como primer paso, se definen el esquema de aprendizaje, el entorno, el estado, las acciones y la función de recompensa. Además, para acelerar el entrenamiento del modelo DQN, se transfiere el conocimiento de otros modelos DQN entrenados en entornos similares con base a un protocolo de evaluación presentado en la Sección 4.3. Posteriormente, en la Sección 4.4, se presentan el mecanismo de gestión del buffer de repetición dual (DER) propuesto para reducir el tiempo transitorio sin demeritar la capacidad de la red durante el entrenamiento de ajuste. Además, se presentan los mecanismos de gestión del buffer de ER implementados para comparar la propuesta del buffer DER. Por último, se definen las métricas para evaluar tanto el tiempo transitorio como la capacidad durante el entrenamiento de ajuste.

4.1 Entorno del esquema de aprendizaje por refuerzo profundo

En el esquema de aprendizaje considerado en esta tesis, el entorno consiste en un sistema celular de N celdas con un canal de acceso múltiple interferente, esto es, todos los

usuarios del sistema comparten el mismo canal de comunicación. En el centro de cada celda se ubica una SBS atendiendo simultáneamente a I_n usuarios en una banda de frecuencia compartida [11], [106], [107], como se muestra en la Figura 4.1(a) y 4.1(b) para un enlace y tres enlaces por celda, respectivamente. Por simplicidad, cada enlace consiste en una antena transmisora $j \in J$ y una antena receptora $i \in I$, de manera que $J = I$.

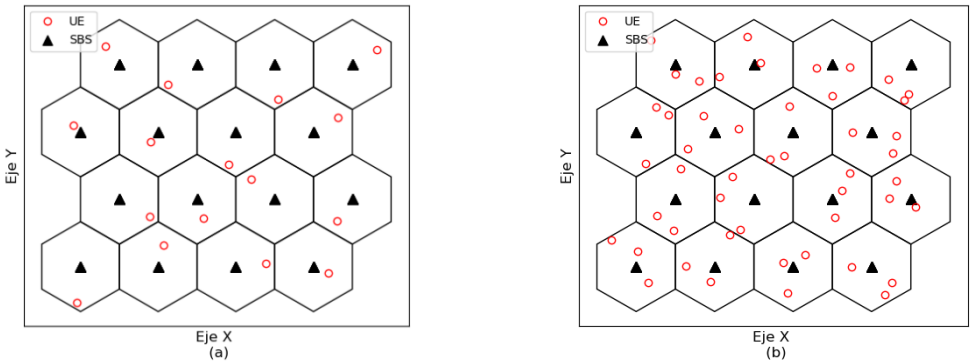

Figura 4.1. Escenario celular con 16 celdas. En cada celda una SBS atiende a un UE. b) En cada celda una SBS atiende a tres UE.

4.1.1 Modelo del canal

Cada antena transmisora es un agente que controla el nivel de potencia de transmisión con base en la información del entorno. Para evaluar la estrategia de control de potencia se consideró un entorno dinámico variante en el tiempo implementando el modelo de canal de Jake [108]. Cada instante de tiempo las condiciones del canal varían aleatoriamente de acuerdo con las condiciones de propagación como lo son el desvanecimiento en pequeña escala y el desvanecimiento a gran escala. El desvanecimiento en pequeña escala se debe a los múltiples caminos que toma la señal transmitida ocasionando que lleguen múltiples copias de la señal en diferente tiempo. Estas señales se combinan en la antena receptora para formar la señal recibida, ocasionando variaciones en la intensidad de la señal. Mientras que el desvanecimiento a gran escala es ocasionado por la distancia que recorre la señal entre el transmisor y el receptor y por los obstáculos (como edificios) existentes entre el enlace de comunicación, causando atenuación en la señal transmitida.

Lo anterior se describe matemáticamente de la siguiente manera. En cada slot de tiempo t, la ganancia independiente entre el transmisor j y el receptor i se denota por:

$$g_{ji}^{t} = \left|h_{ji}^{t}\right|^{2}\beta_{ji}, \tag{11}$$

donde h_{ji}^{t} es una variable aleatoria compleja con envolvente distribuida Rayleigh y β_{ji} es el desvanecimiento por sombreo, representando el desvanecimiento en pequeña escala y a gran escala, respectivamente. De acuerdo con [108], el desvanecimiento a pequeña escala puede ser modelado como un proceso Gauss-Markov complejo de primer orden para cada instante de tiempo consecuente:

$$h_{ji}^{t} = \rho h_{ji}^{t-1} + x_{ji}^{t} \tag{12}$$

donde $x_{ji}^{t} \sim CN(0,1-\rho^{2})$ y ρ es el coeficiente de correlación definido como:

$$\rho = J_{0}(2\pi f_{d} T_{s}) \tag{13}$$

donde J_0 es la función de Bessel de orden cero de primera especie; f_d y T_s son la frecuencia máxima Doppler y el tiempo entre intervalos, respectivamente.

4.1.2 Capacidad de la red

El rendimiento de la red se mide a partir de la Capacidad de Shannon o máxima tasa de datos que se transmiten sin error en un enlace de comunicación. La capacidad de cada enlace i se define como:

$$C_{i} = \log(1+\gamma_{i}^{t}), \tag{14}$$

donde γ_{i}^{t} es el SINR que el enlace i experimenta en la red en el slot de tiempo t. El SINR percibido por el receptor se define como:

$$\gamma_{i}^{t} = \frac{P_{j}^{t} g_{ji}^{t}}{\sum_{j' \neq j}^{J} P_{j'}^{t} g_{j'i}^{t} + \sigma^{2}}, \tag{15}$$

donde P_{j}^{t}es la potencia del transmisor j, g_{ji}^{t} es la ganancia del canal del transmisor j a su receptor i; σ^{2} es la potencia del ruido Gaussiano. El término $\sum_{j' \neq j}^{J} P_{j'}^{t} g_{j'i}^{t}$ representa la interferencia ocasionada por la potencia de transmisión de los transmisores j', incluyendo la interferencia intracelular ocasionada entre los enlaces I_n de cada celda. Lo anterior provoca un sistema limitado por la interferencia y entre mayor sea el número de transmisiones y mayor sea la potencia de transmisión, menor será la calidad (es decir, el SINR) del enlace i.

Por lo tanto, el problema de optimización se establece como la maximización de la suma de capacidades de cada enlace formulada como:

$$\max_{\boldsymbol{P^t}} \sum_{i}^{I} C_i \tag{16}$$

$$s.t. 0 \leq P_j^t \leq P_{max}, \forall j,$$

donde $\boldsymbol{P^t} = \{P_j^t | \forall j\}$ y P_{max} indica la potencia de transmisión máxima.

4.1.3 Eficiencia energética de la red

Además de la capacidad de la red, también se consideró la Eficiencia Energética (EE) como indicador para evaluar el rendimiento del sistema y los efectos de la transferencia del conocimiento en los modelos DQN al considerar diferentes KPI definidos en la sección 4.5. La EE se define como la relación entre la tasa de la capacidad total y la potencia total consumida por unidad de tiempo [109]. Por lo tanto, la EE es expresada como:

$$EE = \frac{\sum_i^I C_i}{\sum_i^I P_i}, \tag{17}$$

Mientras que el problema de maximización de la EE es definido como:

$$\max_{\boldsymbol{P^t}} EE \tag{18}$$

$$s.t. 0 \leq P_j^t \leq P_{max}, \forall j$$

4.2 Esquema de entrenamiento centralizado con ejecución distribuida

En la implementación, los agentes DQN interactúan simultáneamente, por lo que entrenar una política para cada agente DQN ocasiona inestabilidad en el rendimiento del sistema durante el aprendizaje, debido a que los agentes aprenden políticas egoístas enfocándose en maximizar su propia recompensa. En caso contrario, se vuelve complejo entrenar a un agente DQN centralizado que controle el nivel de potencia de cada uno de los transmisores debido a que requiere un diseño de una DNN con más capas o neuronas y un mayor tiempo de entrenamiento. Además, si el número de UE conectados en la red cambia, la capa de salida ya no correspondería al modelo entrenado. Es decir, implementar un modelo centralizado para un entorno de red dinámico requiere entrenar un modelo DQN cada vez

que se produzca un cambio en el entorno o tener múltiples modelos DQN entrenados para un diferente número de UE, lo cual es impráctico para entornos reales. Por lo anterior, similar a los trabajos [11], [107], [110], [111], se implementó el esquema de entrenamiento centralizado con ejecución distribuida (CTDE – centralized training with distributed execution) mostrado en la Figura 4.2. Las ventajas del esquema CTDE respecto al entrenamiento centralizado son: (i) los parámetros de entrada y salida de la DNN son fijos y consideran solo la información de un conjunto de agentes vecinos Z_i y un número fijo de niveles de potencia de transmisión para cada agente i, lo que hace el sistema escalable para cualquier cantidad de UE en la red y (ii) los agentes siguen una política aprendida a partir de las experiencias de todos los gentes, esto permite obtener un proceso de aprendizaje del modelo DQN más estable [112]. A diferencia del esquema DQN centralizado descrito en el Capítulo 3, en el esquema CTDE, cada agente recibe un estado único determinado por Z_i y cada agente realiza una acción independiente, lo que ocasiona que en un instante de tiempo t se generen I experiencias i.e., $e_i^t, \dots, e_I^t$. Este conjunto de experiencias es almacenado en un buffer ER $\mathcal{D}$ localizado en un controlador centralizado (por ejemplo, la MBS). La fase de entrenamiento la ejecuta el controlador centralizado seleccionando un mini-lote de K experiencias del buffer de ER $\mathcal{D}$ para actualizar los parámetros de la DNN. Se envía a todos los agentes una copia de los parámetros actualizados de manera que todos los agentes tengan los mismos parámetros. A pesar de que todos los agentes utilizan la misma DNN para decidir la política de asignación de potencia, cada agente utiliza su información local Z_i, lo que les permite tomar diferentes acciones. Los elementos del MDP para resolver los problemas formulados en (16) y (18) con base a un modelo DQN bajo el esquema CTDE son descritos en la siguiente sección.

4.2.1 Estado

El estado se define como las observaciones que el agente recibe del entorno. En CTDE, cada agente (i.e., antena transmisora) considera al resto de los agentes como parte del entorno. En este sentido, el agente toma la información de Z_i agentes en su vecindad para decidir su política de asignación de potencia.

Sea Z_i el conjunto de enlaces de agentes vecinos cercanos alrededor del agente i. Entonces, el estado de cada agente s_i^t consiste del conjunto de ganancias $\boldsymbol{g}_i^t = \{g_{i',i}^t | i' \in$

$\{i, Z_i\}\}$, el conjunto de potencias de transmisión $\widetilde{\boldsymbol{p}}_i^t = \{P_{i'}^{t-1} \big| i' \in \{i, Z_i\}\}$ y el conjunto de tasa de datos $\boldsymbol{C}_i^t = \{C_{i'}^{t-1} \big| i' \in \{i, Z_i\}\}$ definido como:

$$s_i^t = \{\boldsymbol{g}_i^t, \widetilde{\boldsymbol{p}}_i^t, \boldsymbol{C}_i^t\}. \tag{19}$$

4.2.2 Acción

El modelo DQN requiere un espacio de acción discreto, por lo que el conjunto de acciones $\mathcal{A}$ consiste en l niveles de potencia de transmisión entre una potencia máxima P_{max} y una potencia mínima P_{min}:

$$\mathcal{A} = \left\{0, P_{min}, P_{min}\left(\frac{P_{max}}{P_{min}}\right)^{\frac{1}{|l|-2}}, \dots, P_{max}\right\}. \tag{20}$$

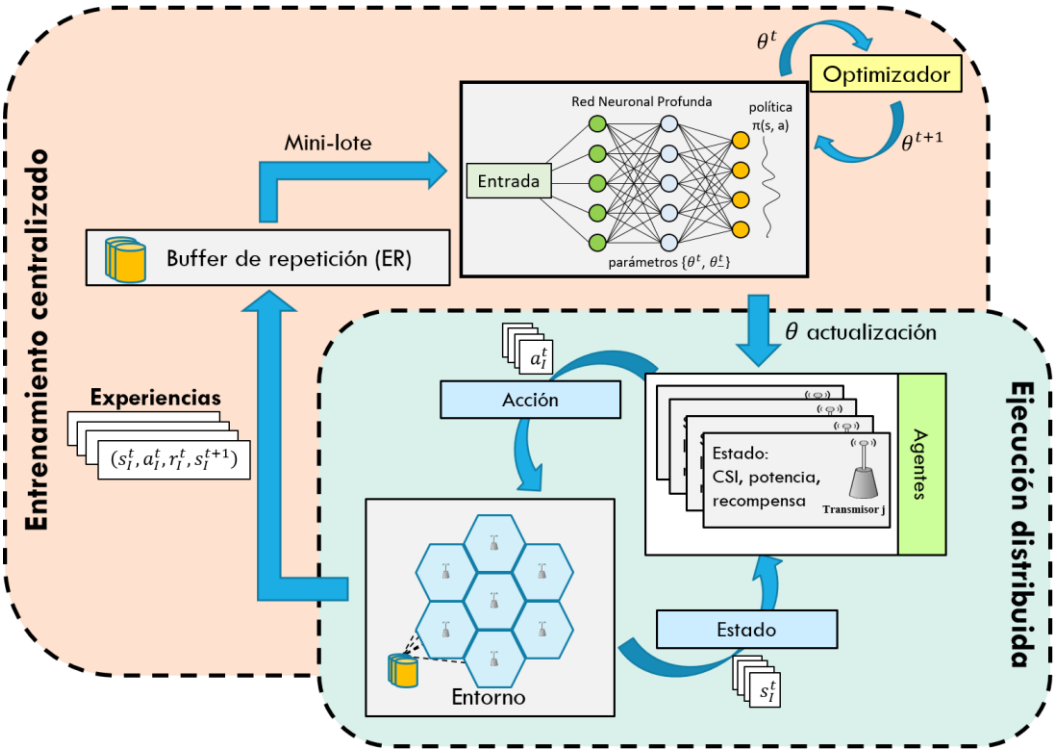

Figura 4.2. Diagrama del esquema de entrenamiento centralizado con ejecución distribuida.

4.2.3 Recompensa

En esta tesis se consideran dos funciones de recompensa: una función de recompensa para maximizar la capacidad de la red y otra función de recompensa para maximizar la EE. Ambas recompensas consideran la interferencia de los agentes vecinos del conjunto Z_i [11].

Para evaluar la función de recompensa basada en la capacidad de la red (i.e., r_c), primero se calcula la capacidad que se obtiene si el transmisor i no está transmitiendo. Es decir, para todos los enlaces I, se calcula la tasa de datos de cada enlace $z \in Z_i$ sin la interferencia causada por el transmisor i:

$$C^t_{z\in Z_i \setminus i} = log_2\left(1 + \frac{P_z g_{z,i}}{\sum_{z' \neq i \in Z_i} P_{z'} g_{z',i} + \sigma^2}\right) \tag{21}$$

Después, el efecto en la capacidad que el enlace i causa al enlace $z \in Z_i$ se calcula utilizando un precio cobrado $\phi^t_{i,z}$ al enlace i por generar interferencia al enlace $z \in Z_i$ [113] definido como:

$$\phi^t_{i,z} = C^t_{z\in Z_i \setminus i} - C^t_z \tag{22}$$

De esta manera la recompensa r_c del agente i en el tiempo t es definida por la capacidad directa del enlace i.e., C^t_i y la penalización por la interferencia causada a sus vecinos i.e., $\sum_{z\in Z_i} \varphi^t_{i,z}$ como:

$$r_{c_i}^{\ t} = C^t_i - \sum_{z\in Z_i} \varphi^t_{i,z} \tag{23}$$

Por otro lado, para evaluar la función de recompensa basada en la EE (i.e., r_e) se implementó una recompensa localizada para medir el rendimiento de la red considerando la EE de cada agente i y la EE del conjunto de vecinos Z_i en el tiempo t. La recompensa r_e del agente i es definida como:

$$r_{e_i}^{\ t} = \frac{C^t_i}{P^t_i} + \frac{\sum_{i' \in D_i} C^t_{i'}}{\sum_{i' \in D_i} P^t_{i'}} \tag{24}$$

4.3 Protocolo de evaluación

Para evaluar el rendimiento de los modelos DQN se diseñó un protocolo de evaluación de tres fases, entrenamiento inicial, transferencia de conocimiento y entrenamiento de ajuste, mostrado en la Figura 4.3.

1) Entrenamiento inicial o Scratch: se divide en U episodios. Al inicio, cada episodio configura aleatoriamente la posición de los UE y el desvanecimiento a gran escala. Las condiciones de pequeña escala se modifican con base a (12) durante T_u intervalos de tiempo. Al final del entrenamiento inicial se genera un modelo base entrenado para diversas condiciones de red. El buffer de ER y los parámetros expertos se almacenan como $\mathcal{D}_0$ y θ_0.
2) Transferencia de conocimiento: Los parámetros DQN y las experiencias del buffer de ER del modelo entrenado en el paso 1 se transfieren (i.e., $\mathcal{D} \leftarrow \mathcal{D}_0$ y $\theta \leftarrow \theta_0$) para iniciar un entrenamiento de ajuste en un entorno de red con condiciones diferentes a los considerados en el entrenamiento inicial.
3) Entrenamiento de ajuste: consiste en V episodios en los que la duración de los intervalos T_v se extiende de manera que el número de experiencias generadas al final del episodio v sean mayores que la capacidad del buffer de ER. Bajo esta condición, al realizar el entrenamiento de ajuste al inicio de cada episodio se elimina el sesgo de retener información de condiciones pasadas en el buffer de ER. Al final de cada episodio se denotan $\mathcal{D}_v$ y θ_v para representar el buffer y los parámetros al entrenar el modelo DQN para la condición del episodio v. Entonces, considerando V episodios, el paso de transferencia de conocimiento se repite al inicio del resto de los episodios, de manera que $\mathcal{D}_v \leftarrow \mathcal{D}_{v-1}$ y $\theta_v \leftarrow \theta_{v-1}$.

A pesar de que el interés en este trabajo de tesis es analizar el rendimiento del modelo DQN durante el entrenamiento de ajuste, este protocolo de evaluación sirve como referencia para mostrar el proceso de generación de diversos parámetros de operación y experiencias, que se reutilizarán durante la fase del entrenamiento de ajuste. Además, el protocolo de evaluación permite controlar los cambios del entorno (diferente número de UE en la red, CSI o localización de UE) arbitrariamente y comparar de mejor forma el rendimiento entre diferentes configuraciones del modelo DQN durante el entrenamiento de ajuste. Es decir, en cada episodio, el modelo DQN aprende una política que maximiza el rendimiento de la nueva

condición. De esta manera, se pueden observar los efectos de aprendizaje durante el entrenamiento de ajuste cuando se utilizan diferentes estrategias de gestión de experiencias consideradas en este trabajo (descritas en las siguientes secciones).

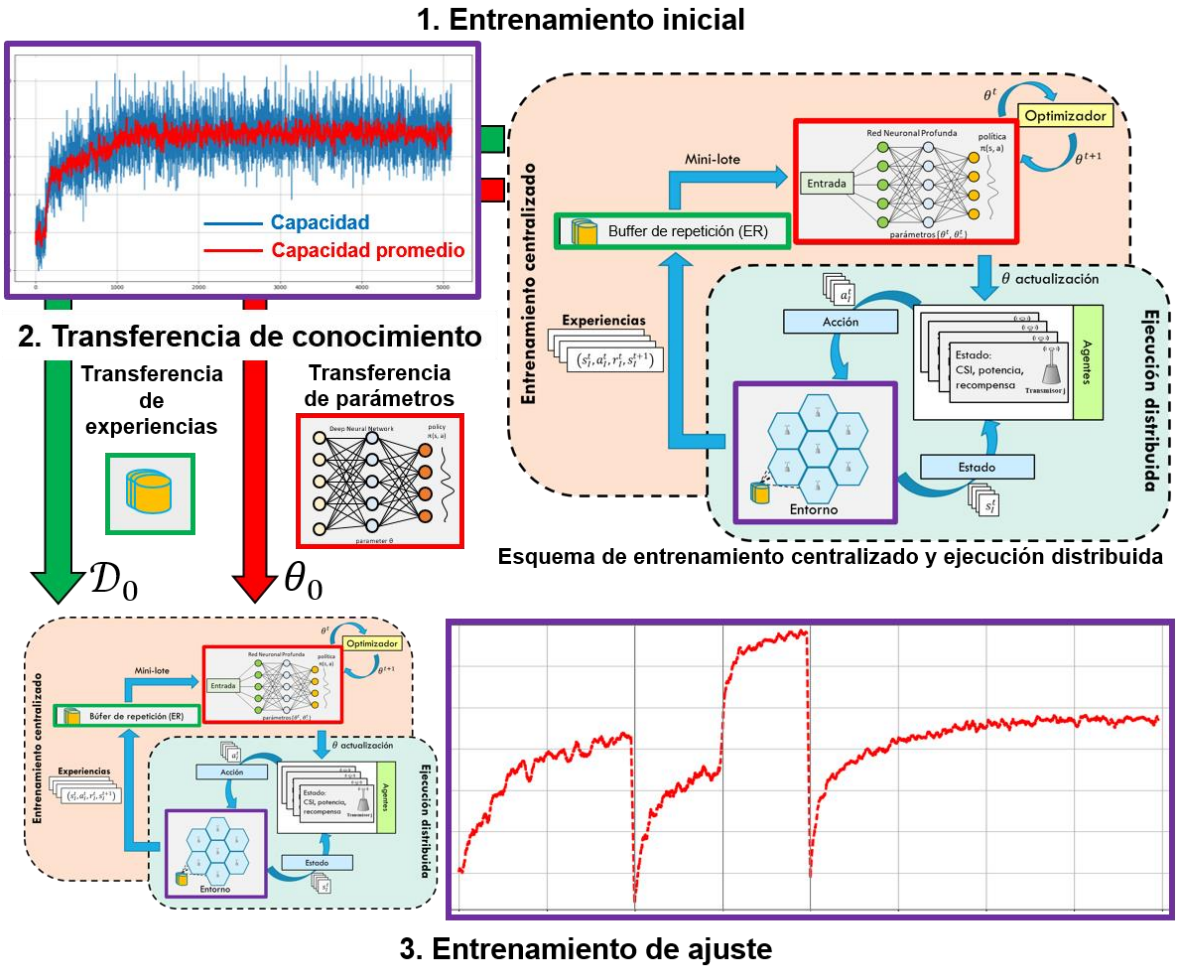

Figura 4.3. Protocolo de evaluación.

4.4 Gestión de experiencias

En esta sección se describen los mecanismos para gestionar las experiencias durante el entrenamiento de ajuste. Se proponen dos mecanismos de gestión de experiencias, el mecanismo de buffer de repetición dual (DER) y una variante para añadir diversidad al buffer de ER denominado mecanismo de repetición de experiencias dual filtradas (FDER – filtered dual experience replay). Ambas estrategias se implementan para reutilizar las experiencias generadas en diferentes entornos de red (i.e., episodios), así como para retener experiencias

que se consideran más relevantes para el proceso de actualización de los parámetros de la DNN durante el entrenamiento de ajuste. Para comparar la eficacia de los mecanismos propuestos, se utilizó como referencia el mecanismo estándar FIFO utilizado en los modelos DQN [20], y dos mecanismos reportados en la literatura, el mecanismo de repetición de experiencias combinadas (CER – combined experience replay) [114] y el mecanismo de experiencias priorizadas (PER – prioritized experience replay) [27]. Además, con base en la transferencia directa TL, se propone un esquema de transferencias instancias para reutilizar las experiencias obtenidas en el paso 1 del protocolo de evaluación descrito en la sección anterior.

4.4.1 Transferencia de instancias

Durante el entrenamiento de los modelos DQN se controlan las condiciones del entorno de red para que los modelos aumenten su capacidad de generalización. En otras palabras, que el modelo DQN logre un rendimiento aceptable bajo cualquier situación que se presente en la red. Sin embargo, entre mayor cantidad de distintas condiciones se consideren durante el entrenamiento mayor será la brecha del rendimiento que el modelo DQN logra en una condición específica y el rendimiento óptimo alcanzable en esta condición. Lo anterior se debe a que a medida que el modelo DQN genera experiencias durante el entrenamiento de ajuste, el modelo comenzará a reemplazar las experiencias antiguas ocasionando un sesgo en el buffer de ER ya que el buffer de ER pierde diversidad (conocimiento de diferentes condiciones) al almacenar solo experiencias de la condición de red actual (i.e., condiciones del episodio). Es decir, cuando se produzca un cambio en el escenario de red y se inicie el entrenamiento de ajuste, es probable que el modelo sea incapaz de seleccionar acciones adecuadas debido al sobreajuste del modelo para una condición de entorno específica. Por lo que, se propone un esquema de transferencia de experiencias experimentadas (EIT – Experienced instance transfer) para romper el sesgo ocasionado por el sobreajuste. La estrategia consiste en reutilizar las experiencias expertas $\mathcal{D}_0$ obtenidas en el entrenamiento inicial cada vez que se inicie un entrenamiento de ajuste como se muestra en la Figura 4.4. También se implementó una transferencia de experiencias directa (DIT – Direct instance transfer). EIT utiliza las experiencias del buffer experimentado cada nuevo episodio v (i.e., $\mathcal{D}_v \leftarrow \mathcal{D}_0$). Mientras que DIT utiliza las experiencias directamente de episodios previos $\mathcal{D}_v \leftarrow$

$\mathcal{D}_{v-1}$. Como punto de referencia también se evaluó el rendimiento al reiniciar el buffer $\mathcal{D}$ cada nuevo episodio, denominado transferencia sin instancias (NIT – Non-instance transfer).

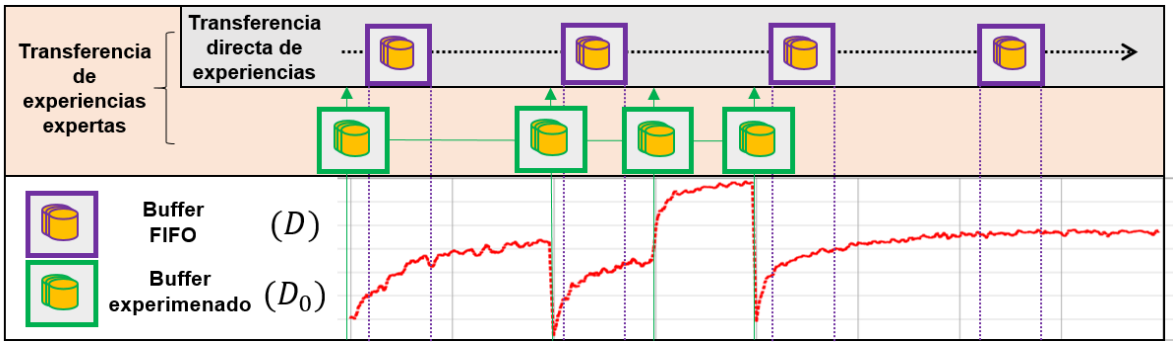

Figura 4.4. Diagrama de esquemas de transferencia de experiencias por EIT y DIT para un escenario que presenta cuatro cambios del entorno durante el entrenamiento.

4.4.2 Repetición de experiencias uniformes (UER)

En este trabajo nos referimos al esquema FIFO como repetición de experiencias uniforme (UER – uniform experience replay). El algoritmo DQN para CTDE con buffer UER es mostrado en el Algoritmo 1. UER es convencionalmente empleado como el mecanismo estándar del modelo DQN para gestionar las experiencias [20]. Bajo el esquema CTDE se generan y almacenan I experiencias en el buffer de ER cada intervalo de tiempo, i.e., $\mathcal{D} \leftarrow \{e_i^t, \dots, e_I^t\}$. Después de cada intervalo de tiempo X se seleccionan K experiencias aleatoriamente para actualizar los parámetros de la DNN. Cuando el buffer de ER alcanza su capacidad B, las experiencias más antiguas se reemplazan por las más recientes.

4.4.3 Repetición de experiencias priorizadas (PER)

El mecanismo de repetición de experiencias priorizadas (PER – prioritized experience replay) consiste en repetir, con mayor frecuencia, experiencias de mayor relevancia para el aprendizaje [27]. La relevancia o prioridad de cada experiencia i es medida con el valor del error TD i.e., $\delta = y^t - Q(s^t, a^t; \theta^t)$, extraído de la función de pérdidas (7), definido como:

$$\wp_i = |\delta_i| + \varpi \tag{25}$$

donde ϖ es un valor positivo bajo para garantizar que el valor de prioridad no sea cero. En PER, cada experiencia tiene un nuevo atributo que indica la prioridad que tiene la experiencia de ser seleccionada. Por ejemplo, la experiencia generada por el agente i en el intervalo t es indicada como $(s_i^t, a_i^t, r_i^t, s_i^{t+1}, \wp_i^t)$. Para asegurar que todas las experiencias sean

Algoritmo 1. Entrenamiento de ajuste del modelo Deep Q-Network con buffer de repetición uniforme

Entrada: Intervalos T_v, intervalo de entrenamiento X, intervalo de actualización L
Salida: Parámetros del modelo DQN entrenados
Inicialización: Inicializar función acción-valor con los parámetros θ_0, inicializar parámetros de la DNN objetivo $\theta_-^t = \theta^t$

Recibir estados iniciales $s_i^t, \dots, s_I^t$
for intervalo $t = 1, \dots, T_v$ **do**
  for $i = 1, \dots, I$ **do**
    Seleccionar acción a_i^t con base en la estrategia ε-codicioso
    if $1 - \varepsilon$
      $a_i^t \leftarrow \max_{a \in A} Q(s_i^t, a; \theta)$
    else
      $a_i^t \leftarrow acción\ aleatoria$
  Ejecutar acciones $a_i^t, \dots a_I^t$, obtener nuevos estados $s_i^{t+1}, \dots s_I^{t+1}$ y calcular recompensas $r_i^t, \dots r_I^t$
  $\mathcal{D} \leftarrow \{(s_i^t, a_i^t, r_i^t, s_i^{t+1}), \dots, (s_I^t, a_I^t, r_I^t, s_I^{t+1})\}$
  Actualizar estados $s_i^t, \dots, s_i^t \leftarrow s_I^{t+1}, \dots, s_I^{t+1}$
  Cada intervalo de entrenamiento X
    Seleccionar mini-lote aleatorio de tamaño K de $\mathcal{D}$
    Calcular paso del gradiente descendente con base en la función (7)
    Actualizar los parámetros de la DNN θ^t utilizando el gradiente calculado
  Cada intervalo de actualización L
    Duplicar parámetros de la DNN objetivo $\theta_-^t = \theta^t$
Devolver los parámetros θ^t y θ^t entrenados

seleccionadas, se les asigna un valor máximo de prioridad igual a 1. Sin embargo, para evitar un sobreajuste por seleccionar las experiencias más recientes se implementa una selección estocástica para asignar una probabilidad de selección a cada experiencia i con base a su valor de prioridad:

$$Probabilidad(i) = \frac{\mathscr{p}_i^{t\,\xi}}{\sum_k \mathscr{p}_i^{t\,\xi}} \tag{26}$$

donde $0 \leq \xi \leq 1$ es un parámetro para ajustar el impacto de la prioridad en la selección de experiencias. Un valor $\xi = 0$ resulta en la selección uniforme del mini-lote, similar a UER. Mientras que un valor $\xi = 1$ resulta en una selección basada totalmente en el valor de prioridad. Utilizar esta selección introduce un sesgo hacia las experiencias con mayor valor de prioridad, lo cual ocasiona un sobreajuste al reutilizar en mayor medida las experiencias con mayor prioridad y sub-utilizar las experiencias con prioridad más baja [27]. Para corregir el sesgo, se utiliza el muestreo de importancia (IS-importance sampling) para calcular el peso de la muestra en el cálculo de la función de pérdida (i.e. $w_i \delta_i$):

$$w_i = \left(\frac{1}{b^t} \cdot \frac{1}{P(i)}\right)^{\varsigma} \tag{27}$$

donde b^t representa el número total de experiencias almacenadas en el buffer en el intervalo t. El parámetro ς se utiliza para controlar la influencia del IS sobre el proceso de entrenamiento. El algoritmo DQN con el mecanismo PER se muestra en el algoritmo 2.

Algoritmo 2: Entrenamiento de ajuste del modelo Deep Q-Network con buffer de repetición priorizada

Entrada: Intervalos T_v, intervalo de entrenamiento X, intervalo de actualización L
Salida: Parámetros del modelo DQN entrenados
Inicialización: Inicializar función acción-valor con los parámetros θ_0, inicializar parámetros de la DNN objetivo $\theta_-^t = \theta^t$

Recibir estados iniciales $s_i^t, \dots, s_I^t$
for intervalo $t = 1, \dots, T_v$ **do**
  for $i = 1, \dots, I$ **do**
    Seleccionar acción a_i^t con base en la estrategia ε-codicioso
    if $1 - \varepsilon$
      $a_i^t \leftarrow \max_{a \in A} Q(s_i^t, a; \theta)$
    else
      $a_i^t \leftarrow acción\ aleatoria$
  Ejecutar acciones $a_i^t, \dots a_I^t$, obtener nuevos estados $s_i^{t+1}, \dots s_I^{t+1}$, calcular recompensas $r_i^t, \dots r_I^t$ y asignar valores de prioridad máxima $\mathcal{p}_i^t, \dots, \mathcal{p}_I^t$
  $\mathcal{D} \leftarrow \{(s_i^t, a_i^t, r_i^t, s_i^{t+1}, \mathcal{p}_i^t), \dots, (s_I^t, a_I^t, r_I^t, s_I^{t+1}, \mathcal{p}_I^t)\}$
  Actualizar estados $s_i^t, \dots, s_i^t \leftarrow s_I^{t+1}, \dots, s_I^{t+1}$
  Cada intervalo de entrenamiento X
    Seleccionar mini-lote aleatorio de tamaño K de $\mathcal{D}$ en base a la probabilidad (26)
    Calcular pesos de IS con base en (27)
    Actualizar prioridad de experiencias seleccionadas
    Calcular paso del gradiente descendente con base en la función (7) utilizando los pesos de IS
    Actualizar los parámetros de la DNN θ^t utilizando el gradiente calculado
  Cada intervalo de actualización L
    Duplicar parámetros de la DNN objetivo $\theta_-^t = \theta^t$
Devolver los parámetros θ^t y θ^t entrenados

4.4.4 Repetición de experiencias combinadas (CER)

Repetición de experiencias combinadas (CER – combined experience replay) es un mecanismo que utiliza la última experiencia del mini-lote cada intervalo de entrenamiento [114]. Similar a PER, las experiencias nuevas tienen mayor prioridad de ser seleccionadas. Sin embargo, PER sigue una estrategia de selección estocástica lo que significa que no garantiza la selección de las experiencias más recientes, mientras que CER garantiza la selección de las experiencias más recientes en el mini-lote. Ya que el protocolo de evaluación considera cambios en el entorno durante el entrenamiento de ajuste, se consideró CER para explorar los efectos en el aprendizaje del modelo DQN de seleccionar las experiencias más recientes que se generan con base en la condición actual del entorno. El algoritmo DQN con

el mecanismo CER se muestra en el algoritmo 3 en el que se selección las I experiencias más recientes y el resto de las $K - I$ experiencias son seleccionadas del buffer $\mathcal{D}$ de forma aleatoria.

Algoritmo 3. Entrenamiento de ajuste del modelo Deep Q-Network con buffer de repetición combinado

Entrada: Intervalos T_v, intervalo de entrenamiento X, intervalo de actualización L
Salida: Parámetros del modelo DQN entrenados
Inicialización: Inicializar función acción-valor con los parámetros θ_0, inicializar parámetros de la DNN objetivo $\theta_-^t = \theta^t$

Recibir estados iniciales $s_i^t, \dots, s_I^t$
for intervalo $t = 1, \dots, T_v$ **do**
  for $i = 1, \dots, I$ **do**
    Seleccionar acción a_i^t con base en la estrategia ε-codicioso
    if $1 - \varepsilon$
      $a_i^t \leftarrow \max_{a \in A} Q(s_i^t, a; \theta)$
    else
      $a_i^t \leftarrow acción\ aleatoria$
  Ejecutar acciones $a_i^t, \dots a_I^t$, obtener nuevos estados $s_i^{t+1}, \dots s_I^{t+1}$ y calcular recompensas $r_i^t, \dots r_I^t$
  $\mathcal{D} \leftarrow \{(s_i^t, a_i^t, r_i^t, s_i^{t+1}), \dots, (s_I^t, a_I^t, r_I^t, s_I^{t+1})\}$
  Actualizar estados $s_i^t, \dots, s_i^t \leftarrow s_I^{t+1}, \dots, s_I^{t+1}$
  Cada intervalo de entrenamiento X
    Seleccionar experiencias $\{(s_i^t, a_i^t, r_i^t, s_i^{t+1}), \dots, (s_I^t, a_I^t, r_I^t, s_I^{t+1})\}$
    Seleccionar mini-lote aleatorio de tamaño $K - I$ de $\mathcal{D} \backslash \{(s_i^t, a_i^t, r_i^t, s_i^{t+1}), \dots, (s_I^t, a_I^t, r_I^t, s_I^{t+1})\}$
    Calcular paso del gradiente descendente con base en la función (7)
    Actualizar los parámetros de la DNN θ^t utilizando el gradiente calculado
  Cada intervalo de actualización L
    Duplicar parámetros de la DNN objetivo $\theta_-^t = \theta^t$
Devolver los parámetros θ^t y θ^t entrenados

4.4.5 Repetición de experiencias dual (DER)

En el esquema CTDE, la cantidad de experiencias almacenadas en cada instante de tiempo crece proporcionalmente con el número de agentes coexistiendo en el entorno, lo que ocasiona que las experiencias permanezcan menor tiempo en el buffer de ER ya que tiene una capacidad limitada. Esta tasa de almacenamiento ocasiona que el buffer de ER se actualice rápidamente y que cada experiencia sea muestreada con menor frecuencia o que incluso no sean muestreadas. Esta situación se vuelve particularmente relevante para el entrenamiento de ajuste en el que las experiencias de exploración suceden con poca frecuencia por la baja probabilidad de exploración. Esta baja probabilidad de exploración se define para disminuir la cantidad de agentes DQN ejecutando acciones aleatorias que podrían afectar el rendimiento de la red. No obstante, estas experiencias describen de mejor manera el comportamiento del nuevo entorno y, por lo tanto, modifican en mayor medida los parámetros de la DNN en el paso de gradiente descendente, ajustando los parámetros hacia una mejor política para adaptarse a la nueva condición del entorno. Para conservar las

experiencias de exploración y mantener diversidad en el espacio de acción en cada mini-lote, se propone retener las experiencias en un buffer de repetición dual (DER – dual experience replay). Las experiencias de exploración y explotación se almacenan en dos buffer de ER diferentes, $\mathcal{B}_e$ y $\mathcal{B}_i$, respectivamente. Las experiencias en $\mathcal{B}_i$ se centran en la política actual, mientras que la implementación de $\mathcal{B}_e$ preserva la diversidad en el espacio de acción al retener las experiencias de exploración (i.e., experiencias de agentes que ejecutan acciones al azar) por más tiempo. Además, la alta tasa de reemplazo del buffer $\mathcal{B}_i$ renueva rápidamente las experiencias generadas por seguir la política anterior lo que permite una mejor adaptación de la política del modelo DQN a las condiciones del nuevo entorno.

Adicionalmente, en cada intervalo de entrenamiento X, se seleccionan τK y $(1-\tau)K$ experiencias de $\mathcal{B}_e$ y $\mathcal{B}_i$, en donde τ es un híper-parámetro para controlar la tasa de selección de cada buffer para formar el mini-lote de K experiencias. El algoritmo DQN para CTDE con buffer DER se muestra en el Algoritmo 4.

Algoritmo 4. Entrenamiento de ajuste del modelo Deep Q-Network con buffer de repetición dual

Entrada: Intervalos T_v, intervalo de entrenamiento X, intervalo de actualización L y tasa de muestreo τ
Salida: Parámetros del modelo DQN entrenados
Inicialización: Inicializar función acción-valor con los parámetros θ_0, inicializar parámetros de la DNN objetivo $\theta_-^t = \theta^t$

Recibir estados iniciales $s_i^t, \dots, s_I^t$
for intervalo $t = 1, \dots, T_v$ **do**
  flag=zeros$[1, \dots N]$
  for $i = 1, \dots, I$ **do**
    Seleccionar acción a_i^t con base en la estrategia ε-codicioso
    if $1-\varepsilon$
      $a_i^t \leftarrow \max_{a \in A} Q(s_i^t, a; \theta)$
    else
      $a_i^t \leftarrow acción\ aleatoria$
      flag[n] =1
  Ejecutar acciones $a_i^t, \dots a_I^t$, obtener nuevos estados $s_i^{t+1}, \dots s_I^{t+1}$ y calcular recompensas $r_i^t, \dots r_I^t$
  for $n = 1, \dots, I$ **do**
    if flag[n]
      $B_e \leftarrow (s_i^t, a_i^t, r_i^t, s_i^{t+1})$
    else
      $B_i \leftarrow (s_i^t, a_i^t, r_i^t, s_i^{t+1})$
  Actualizar estados $s_i^t, \dots, s_i^t \leftarrow s_I^{t+1}, \dots, s_I^{t+1}$
  Cada intervalo de entrenamiento X
    Seleccionar mini-lote aleatorio de tamaño τK de B_e
    Seleccionar mini-lote aleatorio de tamaño $(1-\tau)K$ de B_i
    Calcular paso del gradiente descendente con base en la función (7)
    Actualizar los parámetros de la DNN θ^t utilizando el gradiente calculado
  Cada intervalo de actualización L
    Duplicar parámetros de la DNN objetivo $\theta_-^t = \theta^t$
Devolver los parámetros θ^t y θ^t entrenados

4.4.6 Repetición de experiencias dual filtradas (FDER)

Con el fin de añadir diversidad del espacio de acción y diversidad del espacio de estado, se implementó un mecanismo que combina EIT con DER. Además, para considerar las H experiencias más diversas respecto al espacio de estado del buffer experimentado $\mathcal{D}_0$ se implementó un filtro de diversidad para generar el buffer filtrado $\mathcal{D}_h$ mediante la similitud de coseno [115] definido como:

$$\psi(\mathcal{Q},\mathcal{R}) = \left(\frac{\sum_{s=1}^{\mathcal{S}} \mathcal{Q}_s * \mathcal{R}_s}{\sqrt{\sum_{s=1}^{\mathcal{S}} \mathcal{Q}_s{}^2} * \sqrt{\sum_{s=1}^{\mathcal{S}} \mathcal{R}_s{}^2}} \right) \tag{28}$$

donde $\mathcal{S}$ es el número de elementos del estado s^t en cada experiencia, $\mathcal{Q}$ y $\mathcal{R}$ son dos vectores de estado extraídos para calcular la similitud entre dos experiencias. Una vez generado $\mathcal{D}_h$ se selecciona un mini-lote de K experiencias de los buffers $\mathcal{B}_e$, $\mathcal{B}_e$ y $\mathcal{D}_h$. La tasa de selección de experiencias de $\mathcal{B}_e$ y $\mathcal{D}_h$ se define como τK, mientras que la tasa de selección de experiencias de $\mathcal{B}_i$ como $(1-2\tau)K$. El mecanismo de repetición de experiencias dual filtradas (FDER- filtered dual experience replay) se muestra en el Algoritmo 5.

4.5 Métricas de desempeño

Para medir el rendimiento de los mecanismos de gestión de experiencias UER, PER, CER, DER y FDER, se implementaron métricas basadas en TL [116] como se muestra en la Figura 4.5. La curva roja, denominada curva base, representa el entrenamiento de un modelo DQN sin transferencia o el entrenamiento del modelo DQN entrenado con el mecanismo UER según se indique en el experimento realizado. La curva negra, denominada curva comparativa, representa la curva de entrenamiento utilizando transferencia de conocimiento o algún mecanismo de gestión de experiencias diferente a UER. Adicionalmente, se consideraron las métricas de la capacidad promedio y el rango inter-cuartil (IQR – interquantile range) para complementar la evaluación de los mecanismos de gestión de experiencias, para medir la capacidad la variación en la capacidad del todo el entrenamiento de ajuste. En esta sección, se explican las métricas con base a la capacidad de la red. Sin embargo, las mismas métricas pueden calcularse utilizando la EE de la red.

Algoritmo 5. Entrenamiento de ajuste del modelo Deep Q-Network con buffer de repetición dual filtradas

Entrada: Intervalos T_v, intervalo de entrenamiento X, intervalo de actualización L, tasa de muestreo τ, buffer experimentado filtrado $\mathcal{D}_0$, tamaño del buffer filtrado H
Salida: Parámetros del modelo DQN entrenados
Inicialización: Inicializar función acción-valor con los parámetros θ_0, inicializar parámetros de la DNN objetivo $\theta_-^t = \theta^t$

$index = sort(\psi)[1:H]$ // Seleccionar H experiencias más diversas
$\mathcal{D}_h = \mathcal{D}_0[index]$ //Asignar H experiencias al buffer filtrado
Recibir estados iniciales $s_i^t, \dots, s_I^t$
for intervalo $t = 1, \dots, T_v$ **do**
  flag=zeros$[1, \dots N]$
  for $i = 1, \dots, I$ **do**
    Seleccionar acción a_i^t con base en la estrategia ε-codicioso
    if $1 - \varepsilon$
      $a_i^t \leftarrow \max_{a \in A} Q(s_i^t, a; \theta)$
    else
      $a_i^t \leftarrow acción\ aleatoria$
      flag[n] =1
  Ejecutar acciones $a_i^t, \dots a_I^t$, obtener nuevos estados $s_i^{t+1}, \dots s_I^{t+1}$ y calcular recompensas $r_i^t, \dots r_I^t$
  for $n = 1, \dots, I$ **do**
    if flag[n]
      $B_e \leftarrow (s_i^t, a_i^t, r_i^t, s_i^{t+1})$
    else
      $B_i \leftarrow (s_i^t, a_i^t, r_i^t, s_i^{t+1})$
  Actualizar estados $s_i^t, \dots, s_i^t \leftarrow s_I^{t+1}, \dots, s_I^{t+1}$
  Cada intervalo de entrenamiento X
    Seleccionar mini-lote aleatorio de tamaño τK de $\mathcal{D}_h$
    Seleccionar mini-lote aleatorio de tamaño τK de B_e
    Seleccionar mini-lote aleatorio de tamaño $(1 - 2\tau)K$ de B_i
    Calcular paso del gradiente descendente con base en la función (7)
    Actualizar los parámetros de la DNN θ^t utilizando el gradiente calculado
  Cada intervalo de actualización L
    Duplicar parámetros de la DNN objetivo $\theta_-^t = \theta^t$
Devolver los parámetros θ^t y θ^t entrenados

4.4.1 Capacidad promedio

La capacidad promedio se mide con base a la media aritmética. Es decir, la suma de capacidades individuales de todos los agentes $C^t = \sum_{i=1}^{I} C_i^t$ de cada intervalo t dividida entre el número de intervalos de tiempo totales T calculada como:

$$C^{prom} = \frac{\sum_{t=1}^{T} C^t}{T} \tag{29}$$

4.4.2 Tiempo transitorio

El tiempo transitorio (TT – transient time) mide el tiempo de aprendizaje en intervalos de tiempo requeridos para alcanzar un rendimiento de umbral. El rendimiento de umbral se

define como el rendimiento mínimo o un valor de rendimiento fijo esperado. Sin embargo, ya que, en el dominio de las comunicaciones móviles, el rendimiento alcanzable de la red varía constantemente de acuerdo con las condiciones de propagación y la localización de los UE, se adoptó un valor de umbral de acuerdo con [117]. Una vez terminado el entrenamiento de ajuste, el valor de umbral se establece como el valor de rendimiento equivalente al 90% del valor más alto de la curva de convergencia con del mecanismos de gestión de ER que alcance el mayor rendimiento.

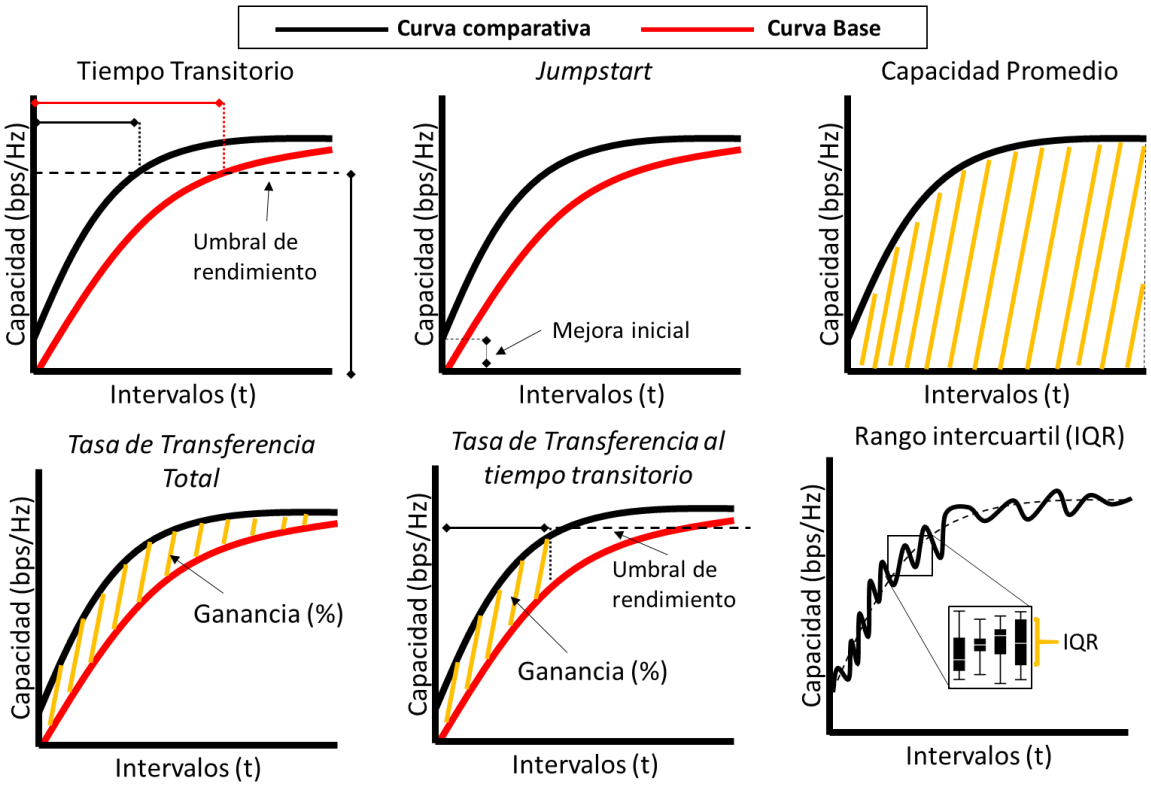

Figura 4.5. Métricas de rendimiento para curvas de entrenamiento individuales y comparación entre curvas de entrenamiento.

4.4.3 Jumpstart

La métrica jumpstart mide el que tan exitoso fue el proceso de transferencia de aprendizaje al iniciar el entrenamiento de ajuste. Esto es, mide el rendimiento de la red inicial entre la transición de episodios y representa la mejora inicial de utilizar un modelo entrenado en un nuevo entorno o episodio. El rendimiento se calcula en el primer slot de tiempo al iniciar un entrenamiento. El valor de la curva comparativa se indica con el subíndice CT, mientras que el valor de la curva base con el subíndice ST. El jumpstart se define como:

$$JS = C_{CT}^{1} - C_{ST}^{1} \tag{30}$$

4.4.4 Tasa de transferencia

La tasa de transferencia (TR – Transfer Rate) mide la ganancia de la capacidad promedio obtenida del entrenamiento con la curva comparativa y el entrenamiento con la curva base, definido como:

$$TR = \left(\frac{C_{CT}^{prom} - C_{ST}^{prom}}{C_{ST}^{prom}} \right) * 100 \tag{31}$$

4.4.5 Tasa de transferencia al tiempo transitorio

En este trabajo se consideró un número fijo de intervalos para el entrenamiento de los modelos DQN. Sin embargo, en la práctica el proceso de entrenamiento se detendrá con base en algún criterio, por ejemplo, cuando el entrenamiento alcance un valor de umbral. Para medir el rendimiento que se obtendría al detener el entrenamiento al alcanzar un valor umbral, se implementó la métrica TR calculando el promedio hasta el tiempo transitorio en lugar de considerar todos los intervalos de tiempo denominada como TRTT (transfer rate until transient time). Las capacidades promedio de las curvas de entrenamiento comparativo y entrenamiento base hasta el tiempo transitorio son indicadas como C_{CT}^{TT} y C_{ST}^{TT}, respectivamente. La tasa de transferencia al tiempo transitorio es calculada como:

$$TRTT = \left(\frac{C_{CT}^{TT} - C_{ST}^{TT}}{C_{ST}^{TT}} \right) * 100 \tag{32}$$

4.4.6 Rango inter-cuartil

Para medir la variación durante el entrenamiento se utilizó una métrica que mide la dispersión entre las diferentes inicializaciones de los modelos [118]. La métrica es el rango inter-cuartil (IQR – interquartile range), el cual mide la diferencia entre los percentiles 75 y 25 o la diferencia entre el cuartil 3 y el cuartil 1. Un menor IQR indica una baja variación del rendimiento durante el entrenamiento, lo que se refleja en una mayor confiabilidad ante comportamientos del rendimiento drásticos (caídas de desempeño) y por lo tanto un entrenamiento más estable.

Capítulo 5

Análisis de resultados

En este capítulo se analizan los resultados obtenidos de la evaluación del entrenamiento de ajuste al implementar el modelo DQN con mecanismo de buffer de Repetición de Experiencias Dual (DER) para resolver el problema de asignación de potencia en una red celular B5G. Se describe el entorno de implementación para entrenar los modelos DQN en una fase inicial y transferir el conocimiento para las evaluaciones del entrenamiento de ajuste en un nuevo entorno (diferentes CSI y localización de UE). Se realizaron cuatro experimentos para evaluar los mecanismos DER, FDER y el esquema de transferencia EIT. En los experimentos se consideran cambios del entorno controlados cada cierto intervalo con el fin de evaluar los puntos de interés (cambio del entorno) con las métricas de tiempo transitorio y capacidad de la red. En el primer experimento, se evalúan los mecanismos UER, CER, PER y DER junto con los esquemas de transferencia EIT, DIT y NIT. En el segundo experimento se evalúa el mecanismo FDER implementado para añadir diversidad en el espacio de acción. A diferencia del primer experimento, en el segundo experimento se consideran los efectos del entrenamiento de ajuste con respecto a las condiciones aprendidas. En el tercer experimento, se considera un solo cambio en el entorno de red, y se evalúan diferentes tamaños del buffer de experiencias, de manera que el modelo almacena menor información del entorno durante el entrenamiento de ajuste. Además, el entrenamiento inicial consiste de una sola condición del entorno en lugar de un entrenamiento con múltiples condiciones como en los primeros dos experimentos. Entrenar el modelo para una sola

condición ocasiona que el modelo DQN ajuste los parámetros específicamente para esta condición, perdiendo adaptabilidad ante nuevas condiciones. Por lo que, bajo las condiciones mencionadas, se evalúa la adaptabilidad del modelo entrenado con diferentes mecanismos de gestión de experiencias (UER, PER, CER y DER) con base a las métricas de tiempo transitorio y capacidad de la red. Por último, se evaluó la EE con diferentes variantes del esquema de transferencias EIT. Cada variante consiste de reutilizar experiencias recolectadas de entornos con menor densidad para evaluar su efecto en el aprendizaje del modelo. Dichos resultados validarán o rechazarán las hipótesis $H1_0$, $H2_0$ y $H3_0$ presentadas.

5.1 Escenario de simulación

En esta sección se describen las características del entorno de red utilizado para ejecutar el entrenamiento inicial y los entrenamientos de ajuste definidos en el protocolo de evaluación del capítulo 4. Esta configuración permite evaluar el rendimiento de los modelos DQN durante el aprendizaje con diferentes mecanismos de gestión de experiencias (UER, PER, UER, CER, DER y FDER).

5.1.1 Configuración del entorno de red

El entorno de la red móvil está formado por 25 celdas de forma hexagonal con un radio de 0.5 km. Cada celda contiene un enlace transmisor-receptor, una antena transmisora localizada en el centro de cada celda y una antena receptora ubicada aleatoriamente dentro de su área de cobertura. Las condiciones del canal de radio se inicializan aleatoriamente, considerando que el desvanecimiento multitrayectoria sigue una distribución Rayleigh con media cero y varianza igual a uno, mientras que el desvanecimiento por sombreo sigue una distribución Gaussiana con media cero y desviación estándar de 8 dB. Las potencias de transmisión máxima y mínima permitidas para los enlaces son de 5 dBm y 38dBm respectivamente. La potencia del ruido es -114 dBm. Las pérdidas por trayectoria se determinan a partir del siguiente modelo $120.9 + 37.6log10(d)$, en donde d es la distancia en km entre cada transmisor y el receptor.

5.1.2 Configuración del modelo DQN

Los parámetros de configuración utilizados para el modelo DQN se muestran en la Tabla 5.1. Se implementó el modelo DQN con UER como el mecanismo de referencia base en los experimentos de esta Tesis. La Figura 5.1 muestra el comportamiento de la capacidad promedio durante el entrenamiento del modelo DQN con UER para diferentes valores de aprendizaje. El modelo logra el mayor rendimiento para los valores de aprendizaje de 0.01 y 0.001. Tratar de encontrar, mediante una búsqueda exhaustiva, el mejor valor de tasa de aprendizaje η (alrededor de 0.001) requiere de más recursos computacionales y el rendimiento no sería significativo. La DNN implementada está formada por dos capas totalmente conectadas con 128 y 64 neuronas con activación ReLU. Las simulaciones se ejecutaron en una máquina con procesador Intel i7-7700 utilizando Python 3.8.3 y Pytorch 1.9.1.

Tabla 5.1. Parámetros del modelo DQN.

Parámetros DQN	Valores
Capas ocultas	2
Conjunto de vecinos	16
Épsilon	0.001
Función de activación	ReLU
Función de pérdida	MSE
Intervalos	10
Intervalos de actualización de la DNN	100
Intervalos de entrenamiento	10
Mini-lote	256
Neuronas	{128, 64}
Optimizador	Adam
Tamaño del buffer	50K
Tasa de aprendizaje	0.001

5.1.3 Configuración del entrenamiento inicial

Como primer paso para la evaluación de los mecanismos de gestión de experiencias se realizó un entrenamiento inicial repetido sobre semillas aleatorias independientes para generar un conjunto de 10 parámetros DNN y 10 buffers de ER experimentados. Por cada entrenamiento se generan 3K episodios (cada episodio representa una condición), donde la posición de los receptores y las condiciones de desvanecimiento se generan de acuerdo a las distribuciones descritas en la sección 4.1.1. Cada condición es generada a partir de una

semilla aleatoria, resultando en una variación del CSI y localización de UE específica para cada condición, de manera que cada condición sea diferente en los episodios del entrenamiento de ajuste. Cada episodio dura 10 intervalos, por lo que el entrenamiento se detiene en el intervalo 30K. El parámetro de exploración se inicializa en 0.9 y se decrementa de forma lineal durante los primeros 10K hasta mantenerse fijo por el resto del entrenamiento en 0.001.

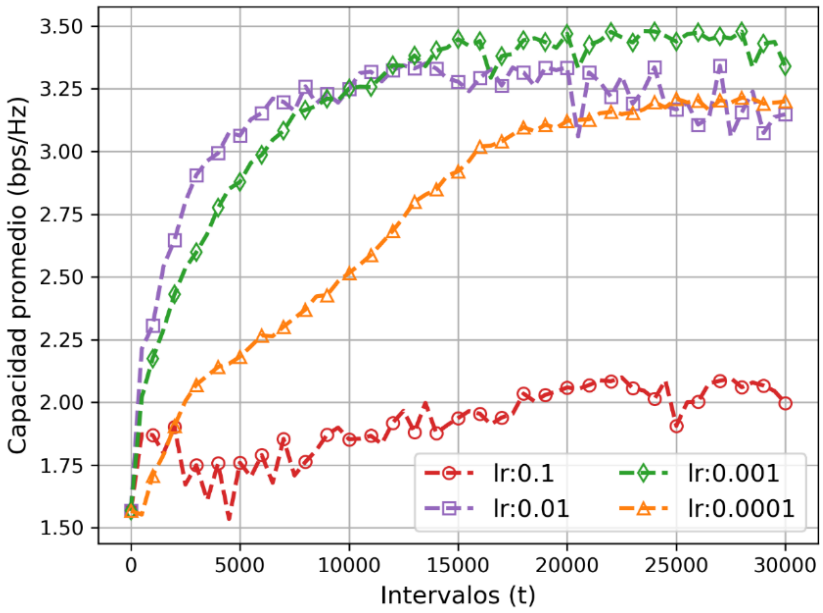

Figura 5.1. Evaluación de la capacidad promedio de la red durante el entrenamiento con diferentes valores de tasa de aprendizaje del modelo DQN con UER en un entorno con 25 celdas y un enlace transmisor receptor en cada celda.

5.1.4 Configuración del buffer de ER

El tamaño de los buffers $\mathcal{B}_e$ y $\mathcal{B}_i$ se configura a 25K cada uno. El híper-parámetro de la tasa de selección (τ) para definir el número de experiencias seleccionadas de los buffers $\mathcal{B}_e$ y $\mathcal{B}_i$ para formar el mini-lote de entrenamiento se define con base en la evaluación de la capacidad promedio para distintos valores de τ mostrados en la Tabla 5.2. La tasa de selección 0.1 logra la mejor capacidad promedio, por lo tanto, las experiencias seleccionadas en cada mini-lote de $\mathcal{B}_i$ y $\mathcal{B}_e$ corresponden a 25 y 231 experiencias, respectivamente.

Tabla 5.2. Capacidad promedio durante el entrenamiento del modelo DQN con DER para diferentes valores de τ.

Tasa de selección (τ)	0	0.1	0.3	0.5	0.7	0.9	1
Capacidad promedio (bps/Hz)	4.1610	**4.33675**	4.31919	4.28423	4.2479	4.16138	3.6910

5.2 Experimento 1: Evaluación de las estrategias de Transferencia de instancias (TI)

En este experimento se evalúa el tiempo transitorio y la capacidad que logra la red al implementar un mecanismo de gestión de experiencias de dos buffers y la reutilización de experiencias mediante una transferencia de instancias con el fin de validar las hipótesis $H1_0$, $H2_0$ y $H3_0$. Transferir instancias se refiere al proceso de transferir experiencias expertas (experiencias generadas por modelos entrenados) entre episodios durante el entrenamiento de ajuste. En este experimento se evalúan tres estrategias de TI; Experienced instance transfer (EIT), Direct instance transfer (DIT) y Non-instance transfer (NIT). Cada estrategia de TI requiere de un mecanismo de gestión de experiencias y, para este caso se evalúan los mecanismos de Repetición Experiencias Uniformes (UER), Repetición Experiencias Priorizadas (PER), Repetición Experiencias Combinadas (CER) y Repetición Experiencias Dual (DER). El indicador de rendimiento que se evalúa es la capacidad que logra la red móvil celular durante el entrenamiento de ajuste del modelo DQN. Este experimento sigue el protocolo de evaluación que consiste en las fases de entrenamiento inicial, transferencia de conocimiento y entrenamiento de ajuste. Primeramente, se consideró la transferencia del buffer y parámetros generados en el entrenamiento inicial (descrito en la Sección 5.1.3). En el proceso de TI, la estrategia EIT reutiliza el buffer experto cada episodio, mientras que la estrategia DIT reutiliza el buffer generado en el episodio anterior. Por otro lado, la estrategia NIT no reutiliza ningún buffer experto, es decir, solo reutiliza las experiencias que genera durante el entrenamiento de ajuste. Por último, las estrategias de TI junto con los mecanismos de gestión de experiencias se evalúan en el escenario del entrenamiento de ajuste.

5.2.1 Configuración del escenario de evaluación

El entrenamiento de ajuste consiste en 8 episodios o condiciones de entorno diferentes con una duración de 5K intervalos de tiempo. La política del modelo DQN se evalúa cada 25 intervalos de tiempo. Para este experimento se implementaron los mecanismos de gestión de

experiencias UER, PER, CER y DER con un buffer de tamaño de 50K. Los exponentes de priorización y del muestreo de importancia para el mecanismo PER permanecen fijos en 0.2 y 0.4 durante el entrenamiento de ajuste debido a que no se diferenciaría de UER si se implementa el decaimiento lineal [27].

5.2.2 Resultados del experimento

La capacidad promedio obtenida durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias se muestran en las Figuras 5.2 y 5.3. En este caso el modelo se evaluó bajo dos condiciones, en un entorno de 25 celdas con un solo enlace (Figura 5.2) y con cuatro enlaces en cada celda (Figura 5.3), respectivamente. En las gráficas, la línea negra punteada representa el rendimiento umbral utilizado para medir el tiempo transitorio de cada curva. Las Figuras 5.2(a), 5.2(c), 5.2(e), 5.3(a), 5.3(c) y 5.3(e) muestran el rendimiento de la red durante cada uno de los ocho episodios. Mientras que las Figuras 5.2(b), 5.2(d), 5.2(f), 5.3(b), 5.3(d) y 5.3(f) muestran el rendimiento promedio de los ocho episodios. Como se muestra en las Figuras 5.2(a)-(f), cada vez que se presenta una nueva condición en la red se puede observar una caída de su desempeño (indicando el jumpstart) ya que el modelo no cuenta con el conocimiento de las nuevas condiciones que se le presentan. Por lo que el modelo DQN requiere ajustar sus parámetros en cada condición.

Para el caso en el que se considera un enlace por celda no se observan diferencias significativas en el rendimiento de los diferentes mecanismos de gestión de experiencias. No obstante, la Figura 5.2(b) muestra una caída de desempeño del 4.54% para el modelo DQN con PER bajo el esquema EIT. Mientras que DQN con CER muestra un tiempo transitorio menor de 1500 intervalos (ver Figura 5.2(d)) resultando en aprendizaje más rápido al reutilizar las experiencias más recientes.

Por otro lado, en la Figura 5.3(b) se muestra que el esquema de transferencia de instancias EIT mejora el rendimiento de todas las curvas hasta un 6.25% y reduce el tiempo transitorio hasta 4K intervalos para el caso en el que se tienen 4 enlaces por celda. El mecanismo DER, en conjunto con los esquemas DIT y NIT, muestra altos valores de capacidad de la red en los primeros 1000 intervalos (ver Figuras 5.3(d) y 5.3(f)). Mientras que el mecanismo PER muestra el menor desempeño bajo los esquemas DIT y NIT. A pesar del bajo rendimiento de PER, este mecanismo se beneficia al reutilizar experiencias antiguas

y se observa en las figuras 5.3(b) y 5.3(d) que se presenta un incremento en el rendimiento de la red del 3.03% y 10.9% con los esquemas DIT y EIT respectivamente en comparación con el esquema NIT. Lo anterior se debe que el esquema PER requiere seleccionar experiencias nuevas con mayor frecuencia para asignarles un valor de prioridad limitando la reutilización de las experiencias con valores de prioridad ya asignados.

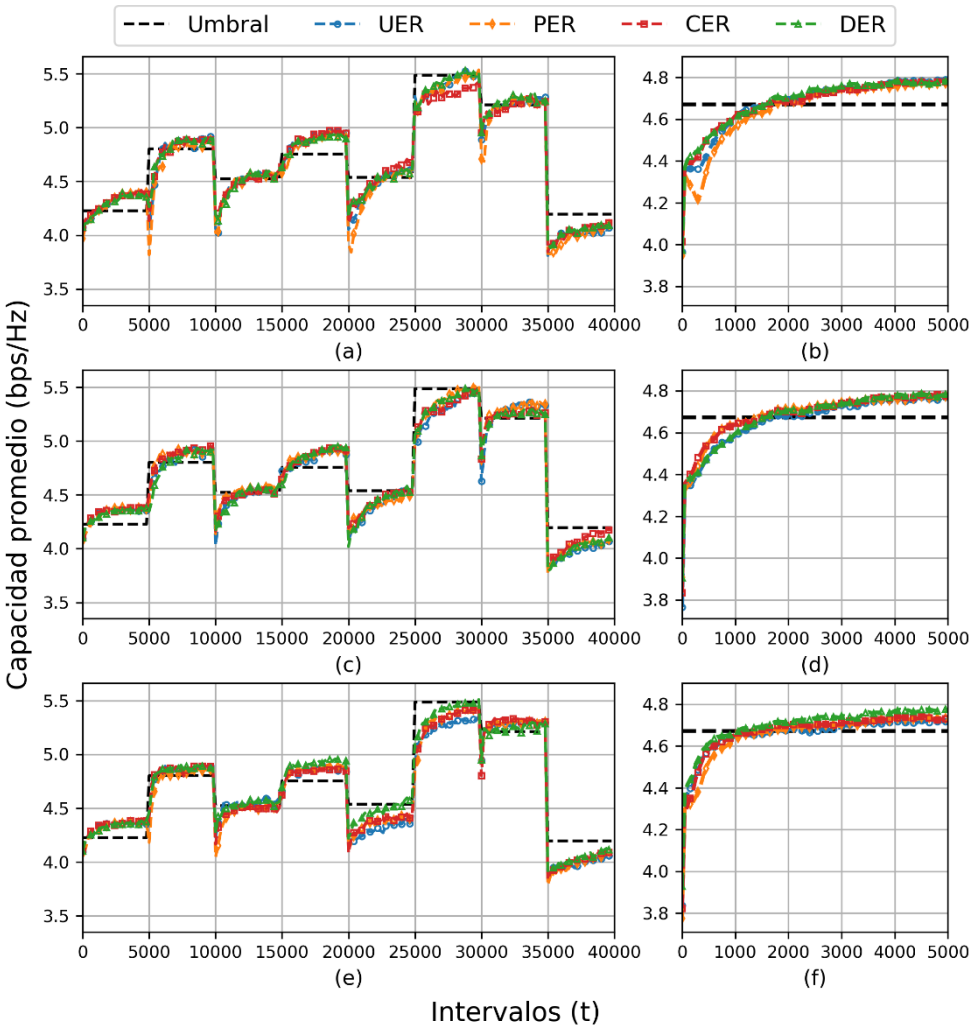

Figura 5.2. Evaluación de la capacidad promedio del modelo DQN con diferentes mecanismos de gestión de experiencias durante el entrenamiento consecutivo de ocho condiciones de entorno para 25 celdas y un enlace por celda. Intervalo de entrenamiento igual 10. (a) Rendimiento por episodio con buffer de ER inicializado con EIT cada episodio. (b) Rendimiento promedio de los episodios con buffer de ER inicializado con EIT cada episodio. (c) Rendimiento por episodio con buffer de ER inicializado con DIT cada episodio. (d) Rendimiento promedio de los episodios con buffer de ER inicializado con DIT cada episodio. (e) Rendimiento por episodio con buffer de ER inicializado con NIT cada episodio. (f) Rendimiento promedio de los episodios con buffer de ER inicializado con NIT cada episodio.

Las Figuras 5.4 y 5.5 muestran la capacidad promedio durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias con un intervalo de entrenamiento de 1 en un entorno de 25 celdas con un solo enlace y cuatro enlaces en cada celda, respectivamente. Al reducir el intervalo de entrenamiento se reduce la cantidad de experiencias generadas antes de actualizar los parámetros. Por otro lado, un valor menor de intervalo de entrenamiento incrementa la reutilización de experiencias almacenadas en el

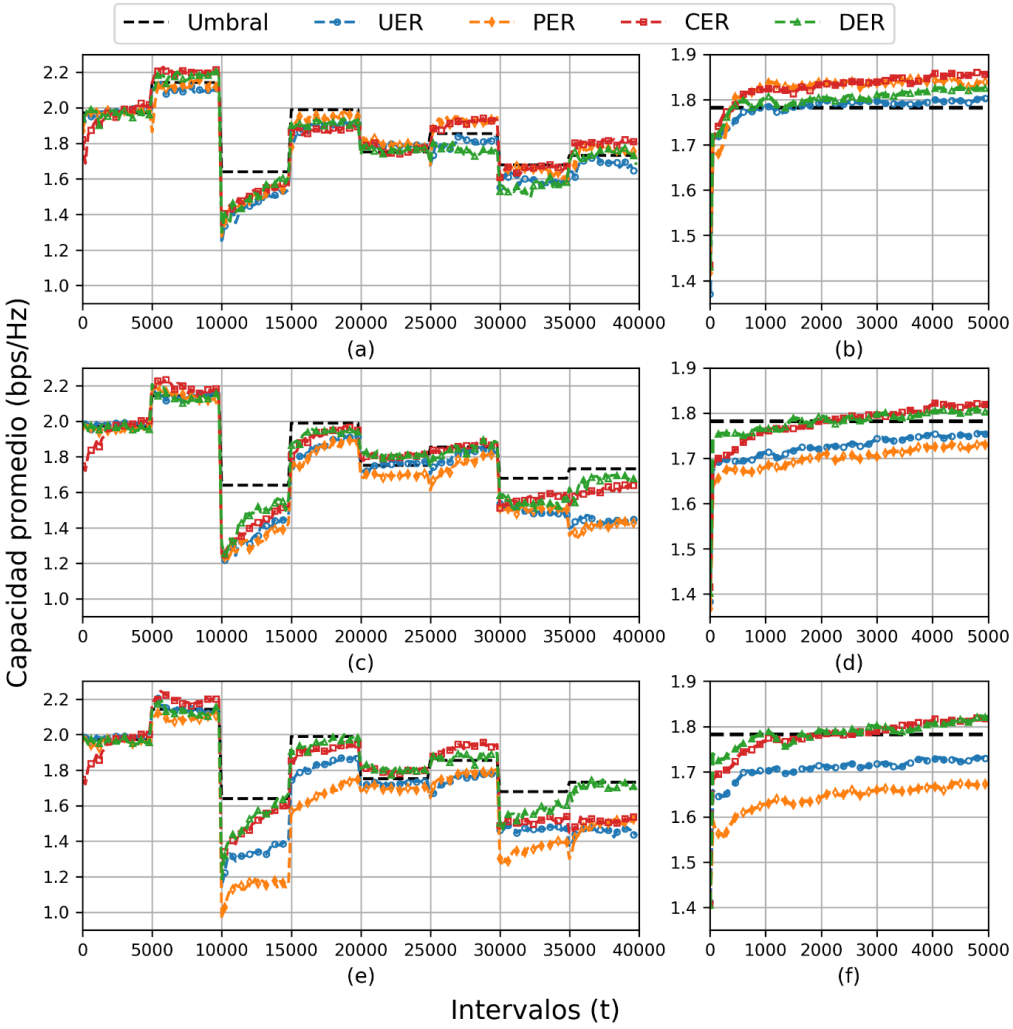

Figura 5.3. Evaluación de la capacidad promedio del modelo DQN con diferentes mecanismos de gestión de experiencias durante el entrenamiento consecutivo de ocho condiciones de entorno para 25 celdas y cuatro enlaces por celda. Intervalo de entrenamiento igual 10. (a) Rendimiento por episodio con buffer de ER inicializado con EIT cada episodio. (b) Rendimiento promedio de los episodios con buffer de ER inicializado con EIT cada episodio. (c) Rendimiento por episodio con buffer de ER inicializado con DIT cada episodio. (d) Rendimiento promedio de los episodios con buffer de ER inicializado con DIT cada episodio. (e) Rendimiento por episodio con buffer de ER inicializado con NIT cada episodio. (f) Rendimiento promedio de los episodios con buffer de ER inicializado con NIT cada episodio.

buffer. Por lo que, se consideró reducir el valor del intervalo de entrenamiento para evaluar su efecto en el rendimiento de la red al implementar los mecanismos de gestión de experiencias UER, PER, CER y DER.

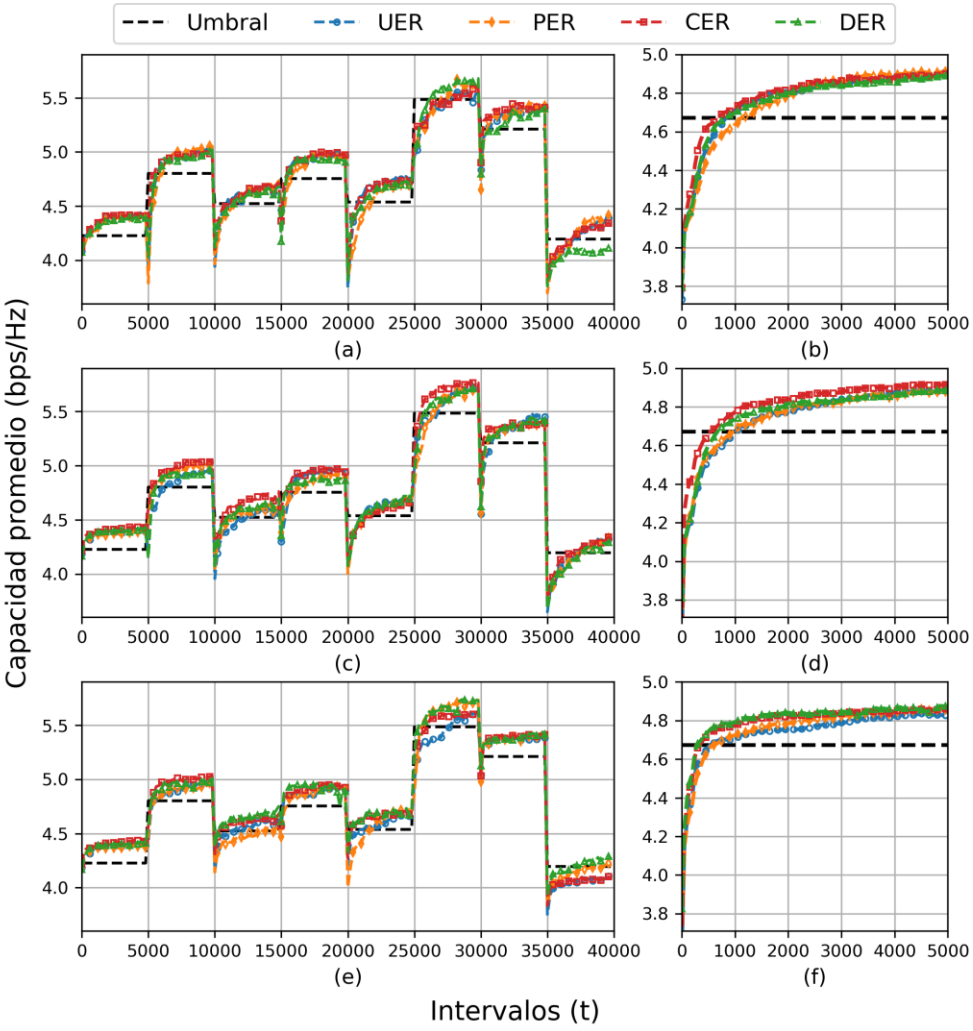

Figura 5.4. Evaluación de la capacidad promedio del modelo DQN con diferentes mecanismos de gestión de experiencias durante el entrenamiento consecutivo de ocho condiciones de entorno para 25 celdas y un enlace por celda. Intervalo de entrenamiento igual 1. (a) Rendimiento por episodio con buffer de ER inicializado con EIT cada episodio. (b) Rendimiento promedio de los episodios con buffer de ER inicializado con EIT cada episodio. (c) Rendimiento por episodio con buffer de ER inicializado con DIT cada episodio. (d) Rendimiento promedio de los episodios con buffer de ER inicializado con DIT cada episodio. (e) Rendimiento por episodio con buffer de ER inicializado con NIT cada episodio. (f) Rendimiento promedio de los episodios con buffer de ER inicializado con NIT cada episodio.

Se puede observar en la Figura 5.4 que el rendimiento de la red es similar al que se muestra en la Figura 5.2 ya que en el entorno de un enlace con celda se percibe menor interferencia y todas las configuraciones encuentran la mejor solución mostrado por sus curvas de entrenamiento superpuestas. Sin embargo, iniciar el entrenamiento de ajuste en un entorno de mayor interferencia (ver Figuras 5.3 y 5.5) muestra una diferencia significativa

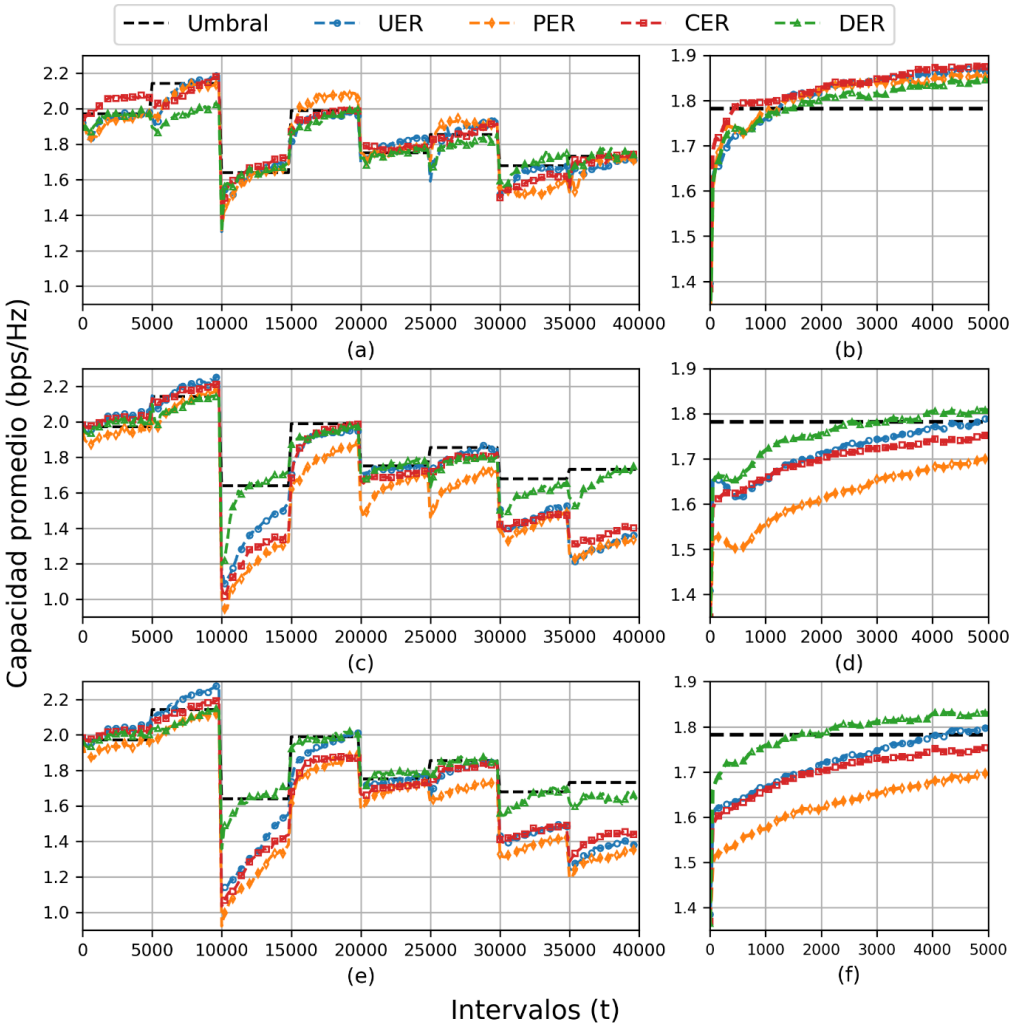

Figura 5.5. Evaluación individual de la capacidad promedio del modelo DQN con diferentes mecanismos de gestión de experiencias durante el entrenamiento consecutivo de ocho condiciones de entorno para 25 celdas y cuatro enlaces por celda. Intervalo de entrenamiento igual 1. (a) Rendimiento por episodio con buffer de ER inicializado con EIT cada episodio. (b) Rendimiento promedio de los episodios con buffer de ER inicializado con EIT cada episodio. (c) Rendimiento por episodio con buffer de ER inicializado con DIT cada episodio. (d) Rendimiento promedio de los episodios con buffer de ER inicializado con DIT cada episodio. (e) Rendimiento por episodio con buffer de ER inicializado con NIT cada episodio. (f) Rendimiento promedio de los episodios con buffer de ER inicializado con NIT cada episodio.

respecto a los mecanismos de gestión implementados. En este caso, implementar el esquema de transferencia de instancias EIT mejora el rendimiento de todas las curvas hasta un 11.58% con una reducción del tiempo transitorio de alrededor de 4K (similar a la reducción de la Figura 5.2(b)). Asimismo, el mecanismo DER muestra una mayor adaptación de las condiciones del entorno bajo los esquemas de transferencia DIT y NIT, los cuales no utilizan el buffer experto (i.e., EIT). Es decir, el entrenamiento de ajuste del modelo DQN con DER alcanza un tiempo transitorio entre 2225 y 1375 intervalos con una tasa de transferencia (TR) de 3.09% y 4.5% respecto al mecanismo UER mostrado en las Figuras 5.5(d) y 5.5(f). Por otra parte, el mecanismo UER supera el rendimiento del mecanismo CER bajo los esquemas DIT y NIT a partir del intervalo 2000 con una ganancia en la capacidad total del 2.11%.

La Tabla 5.3 muestra el concentrado de los resultados obtenidos de los esquemas de transferencia de instancias EIT, DIT y NIT, con los mecanismos de gestión de experiencias UER, PER, CER y DER, para 1 y 4 usuarios (UE) por celda e intervalos de entrenamiento de 1 y 10. Los valores que resaltan en negro representan los mejores resultados obtenidos para cada escenario. Se observa en la Tabla 5.3 que para cualquiera de las estrategias de transferencia de instancias logran su mejor rendimiento con los mecanismos de gestión de experiencias DER y CER independientemente de la cantidad de UE por celda y del intervalo de entrenamiento. Por un lado, DER demuestra un mejor desempeño que el resto de los mecanismos de gestión de experiencias bajo el esquema NIT, con excepción a la métrica JS con intervalos de entrenamiento 1 y 10 para un enlace por celda y la métrica IQR con intervalo de entrenamiento de 1 para uno y cuatro enlaces por celda. Este esquema (NIT) representa una transferencia en un entorno en el cual no se cuenta con información (i.e., experiencias) de antemano. Por otro lado, la combinación CER y EIT muestra que el modelo puede adaptarse más rápidamente a los cambios al reducir el intervalo de entrenamiento a uno, mostrando un mayor rendimiento en la mayoría de las métricas con excepción a la métrica JS para un enlace por celda y la métrica IQR para un enlace por celda. Con la configuración CER con EIT, el modelo puede adaptarse a los cambios si se reutilizan experiencias expertas junto con las experiencias más recientes considerando la diversidad del conocimiento experto y las nuevas condiciones. Sin embargo, para implementar EIT es necesario contar con un buffer experto que contenga las soluciones de entornos similares.

Tabla 5.3. Métricas de desempeño de diferentes esquemas de gestión del entrenamiento del modelo DQN para un entrenamiento consecutivo de ocho condiciones de entorno de red. JS: Jumpstart. TT: Tiempo Transitorio. CP: Capacidad. TR: Tasa de Transferencia. TRTT: Tasa de Transferencia al Tiempo Transitorio. IQR: Rango Inter-cuártil.

Intervalo de entrenamiento: 10

Gestión de experiencias		JS (bps/Hz)		TT (t)		CP (bps/Hz)		TR (%)		TRTT (%)		IQR (bps/Hz)	
		1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE
EIT	UER	4.3360	1.7274	1650	1400	4.6875	1.7879	-	-	-	-	0.1134	0.0876
	PER	4.3426	1.7205	1800	**375**	4.6589	1.8270	-0.6097	2.1863	-1.0290	-3.2833	0.1158	0.0844
	CER	**4.3672**	1.6956	1650	425	4.6647	**1.8322**	-0.4861	**2.4772**	-0.2474	**0.1426**	**0.1063**	0.0786
	DER	4.3545	**1.7317**	**1575**	1350	**4.6916**	1.7961	**0.0875**	0.4593	**0.4096**	-0.1058	0.1086	**0.0778**
DIT	UER	4.2811	1.6640	1575	5000	4.6753	1.7381	-	-	-	-	0.1316	0.1084
	PER	4.3096	1.6305	**1250**	5000	4.6826	1.7031	0.1565	-2.0100	0.5832	-2.0100	0.1317	0.0824
	CER	**4.3121**	1.6660	1525	2625	**4.7158**	1.7833	**0.8660**	2.6007	**1.6751**	2.2314	0.1184	**0.0781**
	DER	4.3054	**1.7447**	1525	**2225**	4.6756	**1.7920**	0.0061	**3.0994**	0.1720	**3.3661**	**0.1181**	0.0814
NIT	UER	**4.3362**	1.6768	2525	5000	4.6886	1.7345	-	-	-	-	0.1067	0.1081
	PER	4.2787	1.6024	2575	5000	4.6397	1.6448	-1.0448	-5.1722	-1.0799	-5.1722	0.1644	0.1369
	CER	4.2541	1.6640	1375	2475	4.6765	1.7830	-0.2585	2.7926	-0.0923	2.0565	0.1152	0.0876
	DER	4.3337	**1.7549**	**1050**	**1375**	**4.7044**	**1.8127**	**0.3366**	**4.5059**	**0.3583**	**4.1405**	**0.1028**	**0.0606**

Intervalo de entrenamiento: 1

Gestión de experiencias		JS (bps/Hz)		TT (t)		CP (bps/Hz)		TR (%)		TRTT (%)		IQR (bps/Hz)	
		1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE	1UE	4UE
EIT	UER	**4.0736**	1.6312	750	1150	4.7826	1.8183	-	-	-	-	0.1143	0.1017
	PER	4.0660	1.5892	1150	1200	4.7691	1.8150	-0.2805	-0.1839	-1.2911	0.8588	0.1239	0.1043
	CER	4.0473	**1.6580**	**650**	**350**	**4.8013**	**1.8335**	**0.3914**	**0.8348**	**2.1709**	**3.4541**	0.1111	**0.0601**
	DER	4.0628	1.6192	800	1200	4.7551	1.8021	-0.5733	-0.8918	0.1017	0.3448	**0.0874**	0.0858
DIT	UER	4.0256	1.6517	1150	4650	4.7634	1.7200	-	-	-	-	0.1030	**0.0721**
	PER	4.0060	1.5207	1150	5000	4.7667	1.6223	0.0699	-5.6778	0.4858	-5.6778	**0.0919**	0.0597
	CER	**4.1595**	1.5791	**500**	5000	**4.8299**	1.7029	**1.3971**	-0.9895	0.7655	-0.9895	0.0985	0.0777
	DER	4.0500	**1.6690**	675	**3150**	4.7769	**1.7970**	0.2834	**4.4813**	**1.1192**	**4.6379**	0.1027	0.0812
NIT	UER	4.0587	1.6081	550	4100	4.7575	1.7273	-	-	-	-	0.0999	**0.0716**
	PER	4.0570	1.4854	600	5000	4.7759	1.6313	0.3869	-5.5571	-0.6317	-5.5571	0.0988	0.0866
	CER	**4.1104**	1.5810	300	5000	4.8089	1.7053	1.0805	-1.2714	1.8814	-1.2714	**0.0773**	0.1040
	DER	4.1061	**1.7005**	**275**	**1625**	**4.8378**	**1.8206**	**1.6864**	**5.4030**	**2.7697**	**7.2361**	0.1038	0.0774

5.2.3. Discusión de resultados

Los resultados de esta sección muestran mejoras del rendimiento del entrenamiento de ajuste al implementar la estrategia EIT con los mecanismos de gestión de experiencias CER y DER en escenarios con uno y cuatro enlaces por celda con intervalos de entrenamiento de 1 y 10. El esquema de transferencias de instancias DIT representa un entrenamiento de ajuste en entornos reales ya que para transferir el conocimiento se requiere conocer los cambios en el entorno de la red, los cuales se controlaron cada 5K intervalos siguiendo el protocolo de evaluación propuesto. Para el caso de 10 intervalos de entrenamiento, se muestra que el modelo DQN acelera su aprendizaje con un TT de 1057 a 1050 intervalos para un enlace por celda y un TT de 1375 a 375 intervalos para cuatro enlaces por celda, al reutilizar experiencias expertas (i.e., esquema EIT) en lugar de no utilizar ninguna experiencia como

en el caso del esquema NIT. Mientras que para el caso de 1 intervalo de entrenamiento, se muestra una reducción del TT de 650 a 275 intervalos con EIT respecto a NIT. Este comportamiento se relaciona con el sesgo del modelo entrenado previamente en el entrenamiento inicial y a la falta de conocimiento por utilizar el esquema NIT. Con el esquema NIT, las experiencias generadas dependen de la política del modelo DQN y de las circunstancias actuales (estado) del entorno. A pesar de que el conocimiento añadido con EIT no contiene las experiencias del entorno del entrenamiento de ajuste, EIT permite ajustar la política del modelo con el conocimiento de otros entornos, generando nuevas experiencias para explorar el entorno sin ejecutar acciones al azar. Además, el esquema EIT mejora el rendimiento general del entrenamiento de ajuste debido a que EIT genera un sesgo al inicio de cada episodio (al cargar el buffer experimentado) para que los modelos DQN reutilicen las mismas experiencias y generen comportamientos similares en todos los mecanismos de gestión de experiencias (UER, CER, PER y DER). Mientras que para los esquemas DIT y NIT, el mecanismo DER influye positivamente en el entrenamiento y aprendizaje del modelo al retener y reutilizar las experiencias de exploración mientras se desconocen las dinámicas del entorno.

5.3 Experimento 2: Evaluación de los mecanismos de Repetición de Experiencias Dual (DER) y Repetición de Experiencias Dual Filtradas (FDER)

En este experimento se evalúa el tiempo transitorio y la capacidad de la red que se obtienen al aplicar los mecanismos de gestión de experiencias DER y FDER los cuales combinan la gestión de experiencias por dos buffers y la reutilización de experiencias previamente adquiridas. Lo anterior, con el fin de validar las hipótesis $H1_0$, $H2_0$ y $H3_0$. Además, en este experimento se analiza la robustez del modelo DQN para mantener el aprendizaje durante varios entrenamientos de ajuste implementando los mecanismos de gestión de experiencias UER, DER y FDER. En cada entrenamiento de ajuste el modelo DQN actualiza sus parámetros de operación para mejorar su rendimiento en la condición del entorno de red actual, lo que ocasiona que el rendimiento en las condiciones aprendidas con anterioridad se reduzca. A diferencia del experimento anterior que se evaluó la política del

modelo en el entorno actual, en este experimento se evalúa cada condición del entorno de red individualmente durante el aprendizaje.

5.3.1 Configuración del escenario de evaluación

En este experimento el escenario de evaluación está formado por 16 celdas, cada una con un enlace transmisor-receptor tanto para el entrenamiento inicial como para el entrenamiento de ajuste. El entrenamiento de ajuste considera 5 condiciones de entorno o episodios generados a partir de cinco semillas aleatorias independientes. Lo anterior genera entornos con diferentes comportamientos de CSI y localización de UE provocando diferentes niveles de interferencia y una capacidad alcanzable distinta en cada episodio. Por lo que para lograr el mejor rendimiento de la red, el modelo requiere ejecutar un entrenamiento de ajuste. Cada episodio dura 3K intervalos. Los parámetros DNN y el buffer de ER se transfieren de forma secuencial entre episodios (i.e., DIT).

Para los mecanismos UER se implementaron dos tamaños de buffer de ER uno de 10K y el otros de 240K (denominados como UER-10K y UER-240K). El buffer de tamaño 240K representa un buffer ilimitado ya que se consideran 16 agentes y 15K intervalos de tiempo generando un total de 240K experiencias al final del entrenamiento. La configuración UER-240K permite analizar los efectos de retención y reutilización de experiencias antiguas respecto a los mecanismos de gestión con buffer de ER limitado. Además, se implementaron los mecanismos de gestión de experiencias DER y FDER con un buffer de exploración y de explotación de tamaño de 5K cada uno. El buffer filtrado utilizado por FDER selecciona, con base al Algoritmo 5, las 10K experiencias de mayor diversidad que se encuentran en el buffer experto de 50K. Por último, se compara el efecto por reutilizar las experiencias del buffer filtrado con UER (denominado FUER) con un tamaño de buffer de ER de 10K.

La evaluación consiste en entrenar un modelo DQN secuencialmente con las 5 condiciones de entorno de red antes mencionadas (similar al experimento 1). Sin embargo, con el fin de evaluar el rendimiento de cada condición durante todo el entrenamiento, se creó un conjunto de evaluación de 50 intervalos de tiempo para cada condición de la red. Después, cada 500 intervalos de tiempo el modelo DQN detiene el proceso de entrenamiento y elige la mejor acción con base a su política actual para obtener el rendimiento del conjunto de evaluación de cada condición. El proceso se repite durante los 15K intervalos de

entrenamiento generando una gráfica de rendimiento para cada condición de entorno, como se muestra en la Figura 5.6.

5.3.2 Resultados del experimento

La figura 5.6 muestra la evaluación de la capacidad promedio para cada una de las cinco condiciones de la red evaluadas. El área en color gris de cada condición de la red indica el intervalo en que el modelo DQN se entrena y evalúa la misma condición de la red. Por consiguiente, la pendiente de convergencia de todas las curvas incrementa en todas las condiciones de la red evaluadas durante los intervalos de tiempo correspondientes a las zonas grises. En estas zonas se observa que la pendiente de la curva UER-240K se reduce cada vez que se presenta una nueva condición en el entorno (i.e., cada 3K intervalos de tiempo). Lo anterior se debe a que el modelo DQN con UER-240K actualiza su política de asignación de potencia con las experiencias de todas las condiciones de red que se presentan, provocando que el modelo se adapte lentamente. Sin embargo, los mecanismos de gestión de experiencias DER y FDER muestran mayor capacidad de adaptación indicado por un tiempo transitorio menor de hasta 66% en los intervalos de la zona gris comparados con UER-10K y UER-240K. Por otra parte, las evaluaciones posteriores a la zona gris muestran el rendimiento del modelo durante el entrenamiento con nuevas condiciones de red. Con excepción a UER-240K, el rendimiento del resto de los mecanismos de gestión de experiencias decae hasta un 26.19%, 13.62%, 18.01% y 10.68% para UER-10K, DER, FUER y FDER, respectivamente con respecto al valor máximo alcanzado en la zona gris (ver Figura 5.6(a)) para la condición 1 en el intervalo 15K. El comportamiento anterior se debe a que el mecanismo UER-240K retiene y utiliza las experiencias antiguas constantemente mitigando el efecto de olvidar el aprendizaje aprendido (la caída de rendimiento es del 2.37%). Mientras que con el mecanismo UER-10K, las experiencias antiguas permanecen menos tiempo en el buffer ya que se reemplazan con mayor rapidez provocando caídas de desempeño del rendimiento cada vez que se presenta una nueva condición de la red. Sin embargo, este descenso del desempeño se amortigua hasta un 16.01% al agregar diversidad al mini-lote (DER, FUER o FDER) con respecto al mecanismo UER-240K, como se muestra en las Figuras 5.6(a) a 5.6(d). Finalmente, las evaluaciones anteriores muestran el rendimiento el modelo DQN evaluado en condiciones que aún no se utilizan en el entrenamiento, por lo que resulta en un bajo

rendimiento para todas las configuraciones. No obstante, utilizar el buffer filtrado en los mecanismos FUER y FDER logra un mayor rendimiento y mayor robustez ante condiciones imprevistas en comparación con UER-240K, UER-10K y DER.

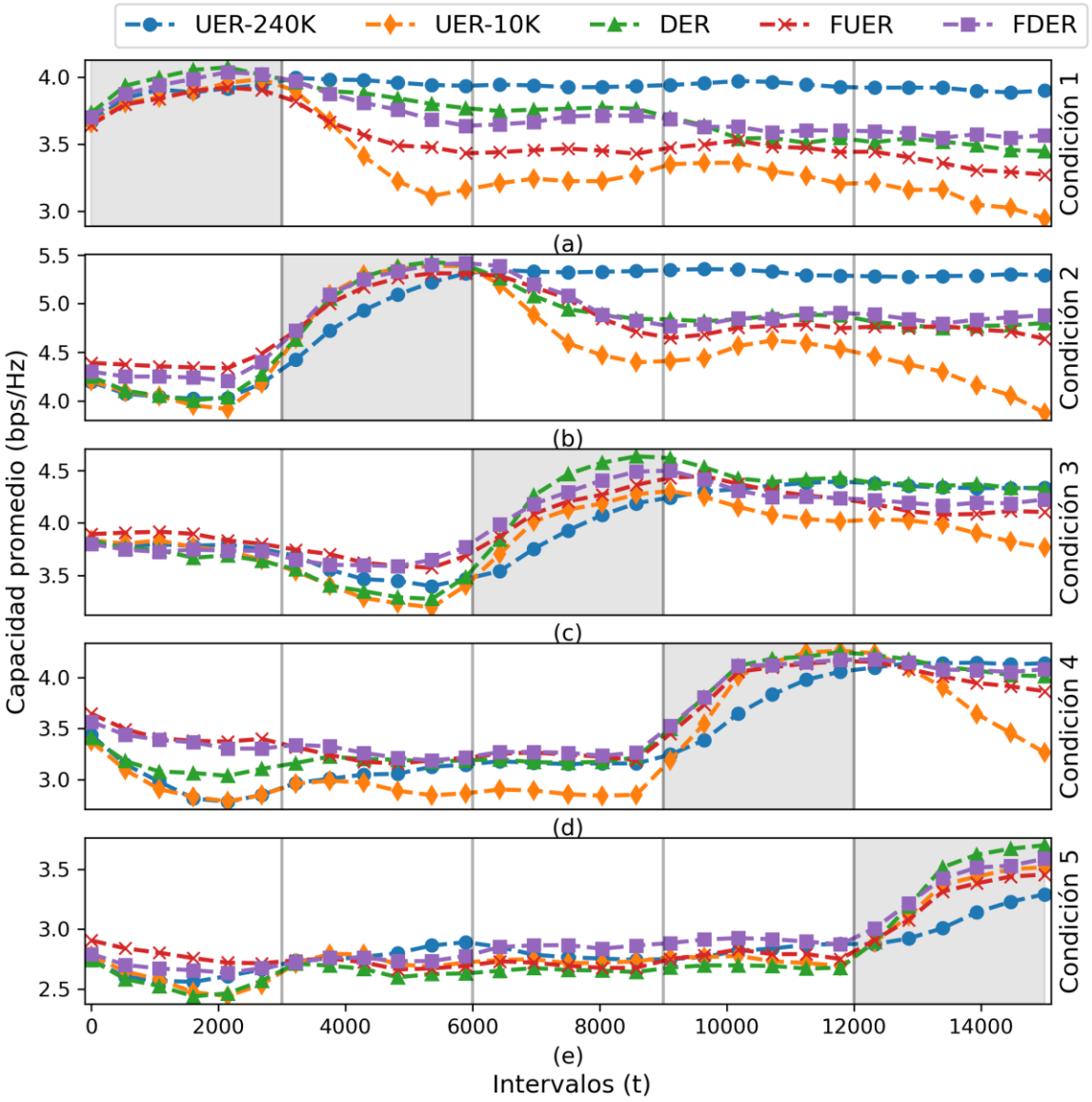

Figura 5.6. Evaluación individual de la capacidad promedio de cada condición de entorno durante el entrenamiento de cinco condiciones consecutivas. Los recuadros en gris indican los intervalos de entrenamiento y evaluación el modelo DQN en la misma condición. (a) Condición 1. (b) Condición 2. (c) Condición 3. (d) Condición 4. (e) Condición 5.

En la Figura 5.7 se presentan las ganancias de capacidad que se obtuvieron a partir de los diferentes mecanismos de gestión de experiencias (UER-10K, DER, FUER, FDER) respecto al modelo DQN transferido al inicio del entrenamiento. El modelo DQN transferido

al inicio del entrenamiento (entrenamiento inicial del protocolo de evaluación) denominado como DQN experto es entrenado bajo diversas condiciones de red (distintas a las condiciones 1-5). En la Figura 5.7(a) se observan las ganancias promedio las capacidades obtenidas a lo largo de los 15K intervalos de entrenamiento de ajuste. Las ganancias son calculadas con respecto al modelo DQN experto. Las ganancias en capacidad alcanzadas durante el entrenamiento comienzan a decaer después de la condición de red-3 para todas las configuraciones. Sin embargo, si se observa la condición de red-5, con FUER y FDER las pérdidas de capacidad son de 3.42% y 1.35% debido a la reutilización del buffer de ER filtrado, mientras que con DER y UER las pérdidas de capacidad son de 4.9% y 4.47%. Por otra parte, en la Figura 5.7(b) se observan las ganancias en rendimiento durante los 3K intervalos de entrenamiento y evaluación de la misma condición (zonas grises en la Fig. 5.6). Estas ganancias en capacidad representan el rendimiento de red alcanzable al ejecutar un entrenamiento de ajuste para una condición específica. En este caso, DER logra una ganancia en capacidad de hasta un 15.15% seguido de FDER, UER-10K y FUER con 13.18%, 12.63% y 12.44%, respectivamente. De las Figuras 5.7(a) y 5.7(b) se puede observar que si el modelo

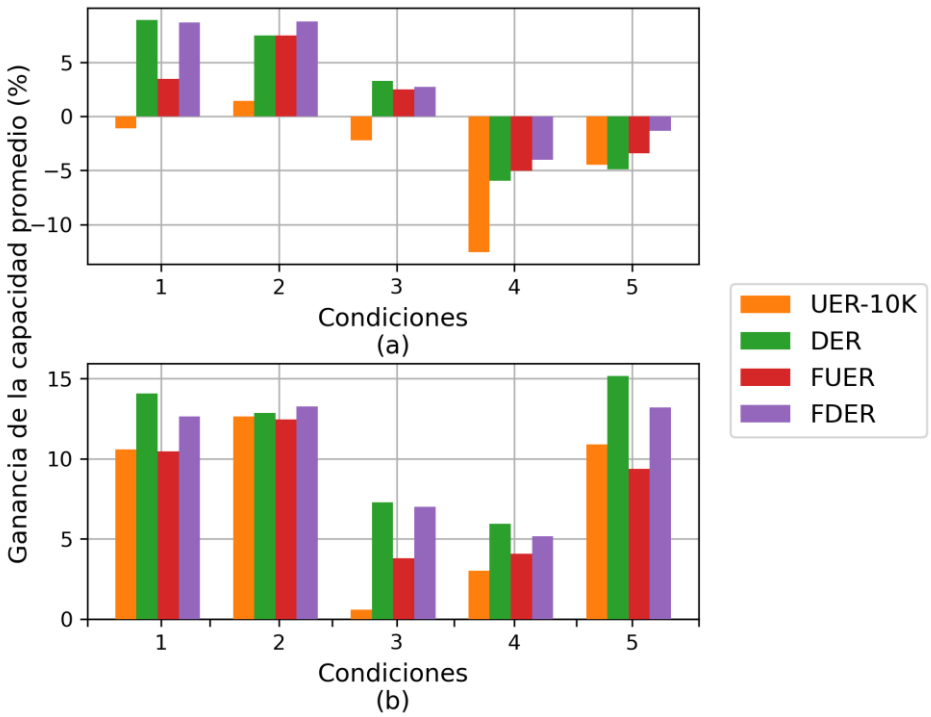

Figura 5.7. Ganancia de la capacidad promedio respecto al modelo DQN experto del entrenamiento inicial. (a) Rendimiento total de la condición. Evaluado durante los 15K intervalos de entrenamiento. (b) Rendimiento específico de la condición. Evaluado durante los 3K intervalos de entrenamiento de cada condición.

DQN se entrena con la gestión de experiencias FDER se obtiene un mejor equilibrio entre adaptabilidad ante nuevas condiciones y robustez para mitigar la caída de rendimiento ante nuevas condiciones de entorno. Mientras que con DER el modelo logra adaptarse mejor a las nuevas condiciones que el resto de las configuraciones logrando una mayor ganancia a costa de menor robustez mostrado en la caída de rendimiento en las condiciones de red-4 y red-5 de la Figura 5.7(a).

En la Figura 5.8 se presenta un comparativo de la diversidad de las experiencias seleccionadas en el mini-lote durante el entrenamiento del modelo DQN con los mecanismos UER-240K, UER-10K, DER, FUER y FDER. Para analizar la diversidad del mini-lote se realizó un mapa de calor que muestra la frecuencia de ocurrencia de las 10 acciones en cada mini-lote. Cada acción representa un nivel de potencia de transmisión definidos en la Sección 4.2.2. La suma de la frecuencia de las acciones en cada intervalo es de 256, igual que el tamaño del mini-lote. Las líneas verticales de color blanco están espaciadas cada 3K intervalos y representan los intervalos límite de cada condición de la red. El color negro en las acciones del mapa de calor representa que el mini-lote seleccionado no cuenta con experiencias ejecutando dicha acción del intervalo correspondiente. Por ejemplo, UER-240K y UER-10K muestran una ausencia de acciones 2-4 en los primeros 500 intervalos. Este comportamiento es similar en las configuraciones de UER ya que utilizan el mismo mecanismo de gestión. Sin embargo, UER-10K remplaza las experiencias más antiguas una vez que el buffer se llena, lo que provoca un comportamiento diferente a partir del intervalo 1500. Este comportamiento se debe a que el tamaño del buffer está limitado, por lo que se ejecutan con mayor frecuencia diferentes acciones con base en la política actual del modelo DQN, las cuales representan las mejores acciones para dicha condición. Por otra parte, los mecanismos DER, FUER y FDER muestran una distribución de la frecuencia de las acciones más equitativas a lo largo del entrenamiento (eje x). A diferencia de UER-10K, la diversidad del espacio de acción mostrada en las Figuras 5.8(c)-5.8(d) no significa que los modelos DQN ejecutan constantemente todas las acciones. Esta distribución en el mapa de calor se debe a la reutilización de experiencias del buffer de exploración y el buffer filtrado, conservando la diversidad en el espacio de acción en cada mini-lote. Esta diversidad de experiencias en el mini-lote permite ajustar la política del modelo para mejorar el rendimiento

el entorno actual. El sobreajuste se evita al considerar experiencias de otras condiciones de red como se mostró en los resultados anteriores.

5.3.2. Discusión de resultados

La primera afirmación de este experimento es que si se incremente el tamaño del buffer y se reutilizan aquellas experiencias relevantes es posible incrementar la adaptación y retención del conocimiento de condiciones previamente entrenadas de los modelos DQN. Sin

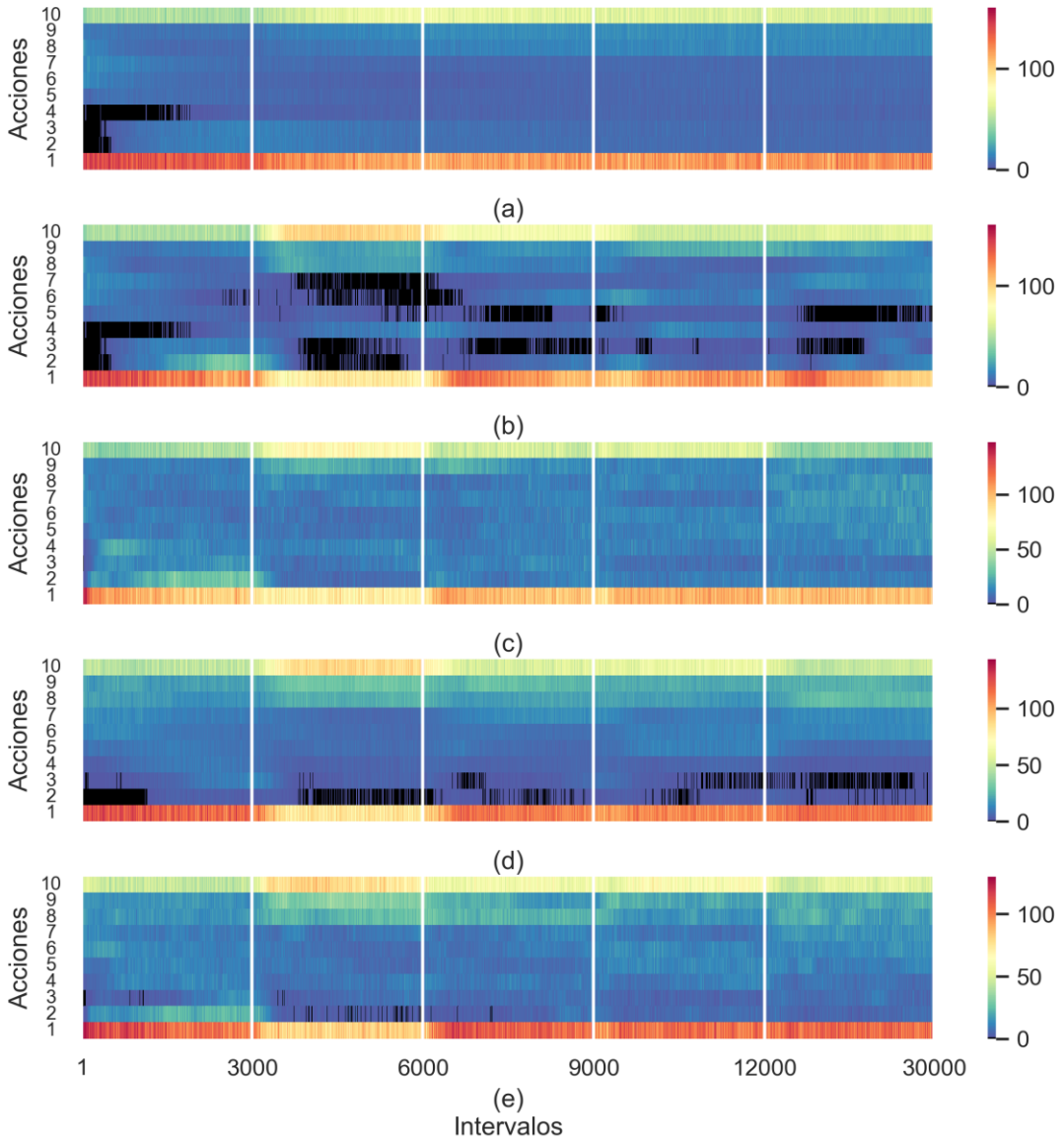

Figura 5.8. Distribución de las acciones seleccionadas en cada mini-lote durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias. (a) UER-240K. (b) UER-10K. (c) DER. (d) FUER. (e) FDER.

embargo, incrementar el tamaño del buffer requiere también incrementar la memoria para los dispositivos que ejecutan los algoritmos DQN. Además, incrementar el buffer no garantiza la diversidad en las experiencias del buffer de ER. A pesar de que los modelos con mayor tamaño del buffer mostraron mayor resiliencia en el rendimiento de condiciones aprendidas con anterioridad, también mostraron menor plasticidad para adaptarse a las nuevas condiciones de red. El diseño de experimentos considera un número fijo de condiciones de red que se mantienen durante 3K intervalos. Sin embargo, al considerar las fluctuaciones del canal que se presentan en los sistemas B5G, la cantidad de condiciones de red consideradas durante el entrenamiento de los modelos DQN y la frecuencia en que estas condiciones de red se mantienen estables producirán un sesgo en la cantidad de experiencias de cada condición. Como muestra la Figura 5.6, el entrenamiento del modelo DQN se acelera si cuenta con diversidad en el buffer de ER o diversidad en los mini-lotes de entrenamiento implementando FDER o DER. Por lo que, con el fin de ajustar los modelos DQN en sistemas reales es recomendable implementar mecanismos para preservar la diversidad de las experiencias del buffer o tener un buffer auxiliar (como en el esquema EIT) con diversas experiencias que sean seleccionadas de acuerdo con las condiciones que se presenten durante la operación de la red móvil.

5.4 Experimento 3: Evaluación de los mecanismos de gestión de experiencias durante el entrenamiento de ajuste

En este experimento se evalúa el tiempo transitorio y la capacidad de la red al implementar un mecanismo de gestión de experiencias de dos buffers independientes (exploración y explotación) con el fin de validar las hipótesis $H2_0$ y $H3_0$. Además, se analiza el efecto por utilizar diferentes tamaños de buffer en diferentes mecanismos de gestión de experiencias sobre el rendimiento de la red al ejecutar un entrenamiento de ajuste. A diferencia del experimento 1, en este experimento el entrenamiento inicial se ejecuta con una sola condición de red, esto ocasiona que el modelo requiera de un entrenamiento de ajuste más prolongado para adaptarse a nuevas y diferentes condiciones de red.

5.4.1 Configuración del escenario de evaluación

El escenario de evaluación consiste en 25 celdas con un enlace transmisor-receptor en cada celda para el entrenamiento inicial. El entrenamiento inicial se realizó por 30K intervalos en una sola condición de red. En este caso solo se consideró la transferencia de parámetros DNN expertos (i.e., NIT). El entrenamiento de ajuste se realizó durante 30K intervalos para una nueva condición de red diferente a la que se utilizó durante el entrenamiento inicial. El objetivo de este experimento es evaluar la capacidad del modelo DQN para adaptarse a nuevas condiciones de red. El esquema consiste en transferir un modelo sobre ajustado a una sola condición de red. Este sobre ajuste provocará que algunas acciones se ejecuten con mayor frecuencia ocasionando un sesgo en la diversidad de las experiencias del buffer de ER al ejecutar constantemente acciones adecuadas para el entorno del entrenamiento inicial y no necesariamente para la condición de red actual.

En esta evaluación se consideraron los mecanismos de gestión de experiencias UER, CER, PER y DER con tamaños de buffer de 10K y de 50K. Se implementó el modelo UER entrenado con parámetros inicializados al azar y sin transferencia de buffers denominado *Scratch*, para utilizarlo como referencia.

5.4.2 Resultados del experimento

Las Figuras 5.9 y 5.10 muestran la capacidad promedio durante el entrenamiento del modelo DQN para diferentes mecanismos de gestión de experiencias en un entorno de red celular con 25 celdas con un enlace y cuatro enlaces por cada celda, respectivamente.

La Figura 5.9(a) muestra la capacidad promedio del entrenamiento al utilizar un tamaño de buffer de 10K. El modelo DQN muestra una convergencia más rápida con un tiempo transitorio de 1000 intervalos de tiempo cuando se entrena con el mecanismo DER. Por otro lado, el rendimiento de UER, PER y CER muestran una capacidad promedio de convergencia de entre 4.1-4.34 bps/Hz y un tiempo transitorio de entre 4500 a 5500 intervalos. La Figura 5.9(b) muestra que el incremento del buffer a 50K mejora la capacidad promedio y el tiempo transitorio de todos los mecanismos de gestión de experiencias. En este caso, el entrenamiento desde cero o *Scratch* logra una mayor capacidad final mostrando que el sesgo por transferir los parámetros del modelo DQN inicial y por la falta de exploración limita el rendimiento alcanzable de la condición del entorno actual. Este comportamiento es

similar al observado en la sección 5.2.2, en el que la pendiente de las curvas de convergencia en los intervalos 30000 a 40000 de las configuraciones UER y PER en la Figura 5.3(c) y 5.3(e) se reduce o se mantiene horizontal, lo que significa que el modelo se quedó atrapado en un óptimo local.

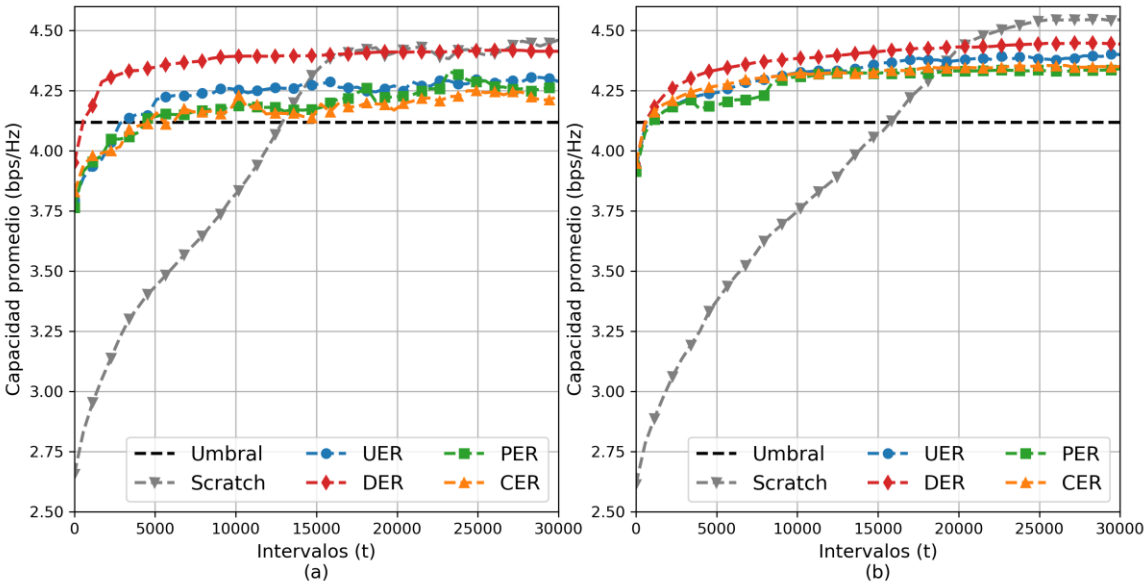

Figura 5.9. Evaluación de la capacidad promedio de la red durante el entrenamiento de ajuste bajo diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y un enlace transmisor-receptor en cada celda. (a) Tamaño del buffer de 10K. (b) Tamaño del buffer de 50K.

La Figura 5.10 muestra la capacidad promedio que logra la red en un entorno de alta interferencia (cuatro enlaces por celda) cuando se transfiere el modelo DQN entrenado previamente en una condición de red con poca interferencia (un enlace por celda). Comparado con los resultados de la Figura 5.9, se observa una menor pendiente de convergencia. A excepción del mecanismo de gestión de experiencias DER, el resto de las técnicas (UER, PER y CER) no logran converger más allá del umbral preestablecido. Incluso, el mecanismo CER, con tamaño de buffer de 10K, comienza a divergir a partir del intervalo 14K como se muestra en la Figura 5.10(a). En la Figura 5.10(b) se puede observar que cuando se incrementa el tamaño del buffer a 50K también se incrementa el rendimiento de los modelos DQN. No obstante, la inestabilidad que presentan los mecanismos de gestión UER, PER y CER oscila entre un 1.42 y 1.7 bps/Hz a partir del intervalo 8500, lo que significa que estos mecanismos no son adecuados para un entrenamiento de ajuste ya que resultarán en un rendimiento de red impredecible.

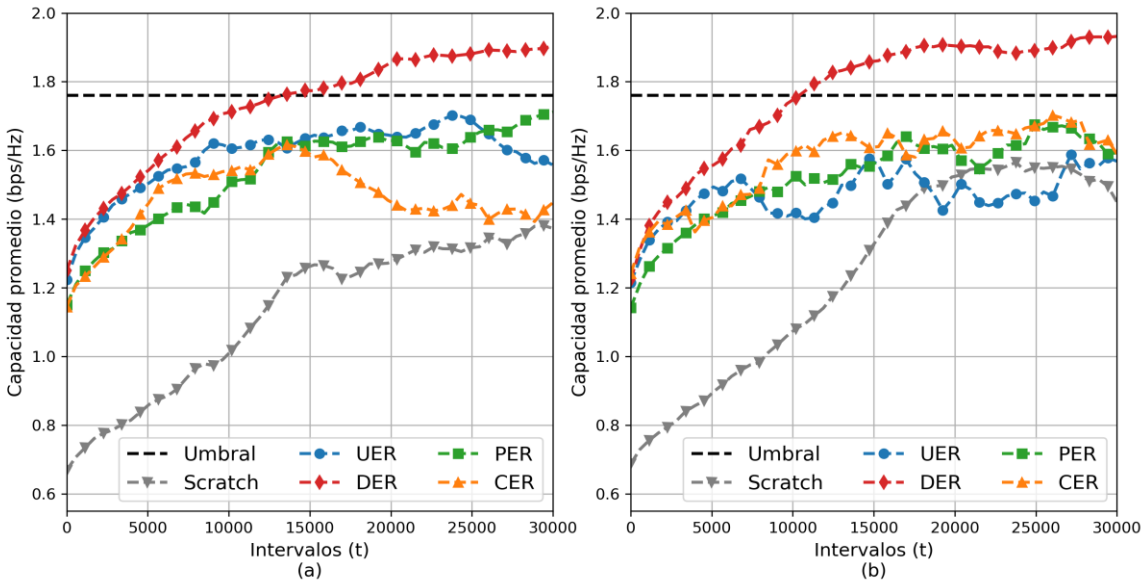

Fig. 5.10. Evaluación de la capacidad promedio de la red durante el entrenamiento de ajuste bajo diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y cuatro enlaces transmisor-receptor en cada celda. (a) Tamaño del buffer de 10K. (b) Tamaño del buffer de 50K.

La Figura 5.11 muestra la variación de la capacidad promedio durante el entrenamiento del modelo DQN cuando se tiene un solo enlace por celda. La gráfica muestra los resultados de un entrenamiento con UER, PER y CER y un buffer de tamaño de 10K, el resultado del mecanismo DER se muestra para propósitos de comparación en las Figuras 5.11(a), 5.11(b) y 5.11(c), respectivamente. El mecanismo DER presenta la menor variación de capacidad de la red, mientras que los mecanismos UER, PER y CER, muestran desvanecimientos del rendimiento de hasta por 13.95% con respecto al rendimiento obtenido por el mecanismo DER. No obstante, UER y PER muestran algunos aumentos súbitos que rebasan hasta en un 6.34% al rendimiento obtenido por DER. A pesar de la presencia de estos picos de capacidad de la red, el entrenamiento con UER, PER y CER es inestable entre intervalos provocando mayor incertidumbre en el rendimiento esperado en comparación cuando se ejecuta el entrenamiento con el mecanismo DER.

En la Figura 5.12 se muestra las variaciones de la capacidad de red al incrementar el tamaño del buffer a 50K (se consideró la misma configuración de 25 celdas y un solo enlace por celda que para la Figura 5.11). Se puede observar que cuando se incrementa el tamaño del buffer se reducen los decrementos y aumentos súbitos de la capacidad (aquellos

mostrados en la Figura 5.11). La magnitud de los desvanecimientos se reducen 4.65%, sin embargo, se requiere un incremento en el tamaño del buffer del 400%.

La Figura 5.13 muestra las variaciones de la capacidad de red para un entrenamiento del modelo DQN con un tamaño de buffer de 10K y un entorno de red de 25 celdas y cuatro enlaces por celda. En este caso se consideró la transferencia del modelo DQN entrenado con un enlace (1-UE) al modelo DQN entrenado con cuatro enlaces (4-UE). Se puede observar en las gráficas que los modelos de gestión de experiencias presentan un comportamiento de mayor inestabilidad con respecto a los resultados mostrados en las Figuras 5.11 y 5.12. La variación de la capacidad promedio oscila de entre 1.8 y 0.6 bps/Hz, el mayor decremento en

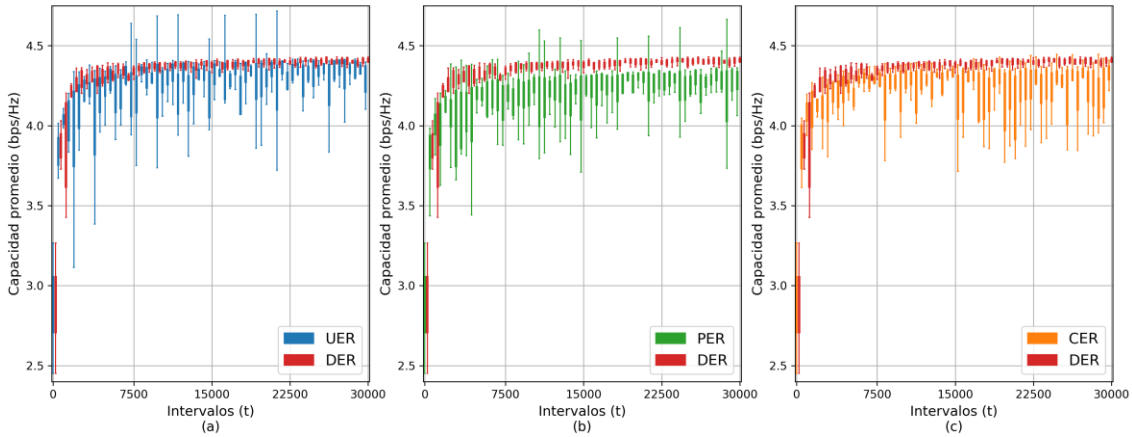

Figura 5.11. Variación de la capacidad promedio de la red durante el entrenamiento del modelo DQN con DER y un buffer de 10K respecto a diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y un enlace transmisor-receptor en cada celda. (a) UER. (b) PER. (b) CER.

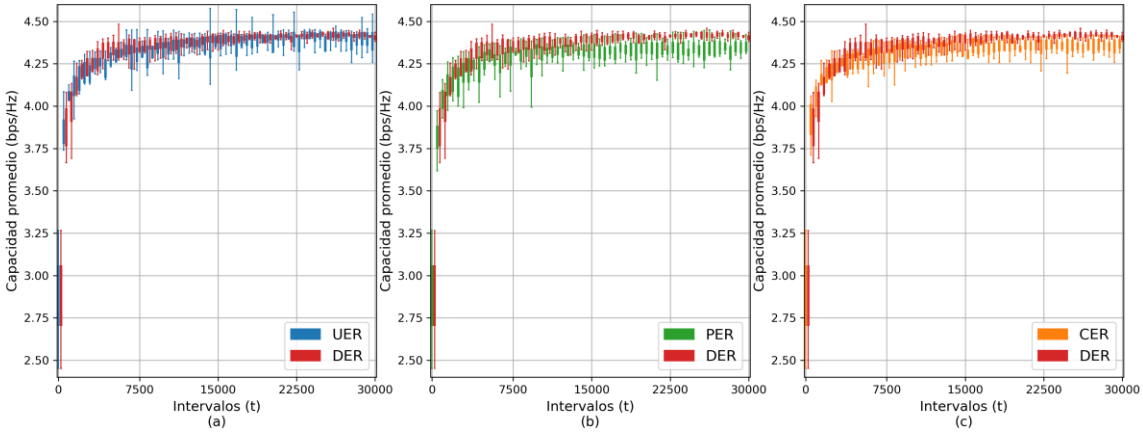

Figura 5.12. Variación de la capacidad promedio de la red durante el entrenamiento del modelo DQN con DER y un buffer de 50K respecto a diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y un enlace transmisor-receptor en cada celda. (a) UER. (b) PER. (b) CER.

capacidad es del 64.7% con respecto al mecanismo DER. Todos los mecanismos de gestión de experiencias (UER, PER y CER) se comportan de manera similar en los primeros 7K intervalos debido a que el ajuste de los parámetros se activa una vez que las experiencias almacenadas son mayores al tamaño del mini-lote, que es de 256. Esto provoca un ajuste similar de los parámetros de la DNN al inicio del entrenamiento. A medida que el entrenamiento de ajuste progresa, las experiencias generadas por las acciones de los agentes se gestionan de forma diferente siguiendo los mecanismos UER, PER, CER y DER provocando diferentes comportamientos después del intervalo 7K.

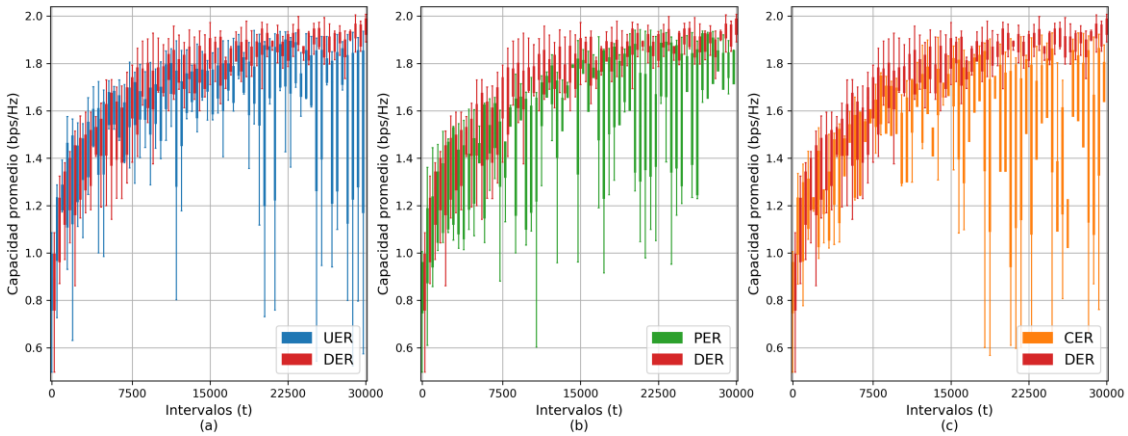

Figura 5.13. Variación de la capacidad promedio de la red durante el entrenamiento del modelo DQN con DER y un buffer de 10K respecto a diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y cuatro enlaces transmisor-receptor en cada celda. (a) UER. (b) PER. (b) CER.

La Figura 5.14 muestra la variación de la capacidad para un entrenamiento del modelo DQN con un tamaño de buffer de 50K y un entorno con cuatro enlaces en cada celda. En este caso, el incremento del buffer de los mecanismos de gestión UER, PER y DER reduce los picos de las caídas de desempeño en un 45% con respecto a DER. Además, se observa una reducción de las variaciones de la capacidad promedio del mecanismo DER, oscilando entre 1.88 y 1.99 bps/Hz al final del entrenamiento. A diferencia de la Figura 5.12, el incremento en el tamaño del buffer reduce en menor medida la inestabilidad del entrenamiento para los mecanismos de gestión UER, PER y CER. Lo anterior se debe a que las curvas de entrenamiento de la Figura 5.14 no han encontrado la mejor solución, mostrado por su variación en la capacidad de la red. Incrementar el tamaño permite retener experiencias antiguas y por lo tanto, reutilizarlas en cada selección del mini-lote, incrementando su diversidad y estabilizando su entrenamiento de ajuste mostrado en la Figura 5.12. Sin

embargo, en la Figura 5.14, el entorno es más interferente y el número de experiencias generadas es mayor (por el número de agentes) dificultando encontrar la mejor solución y reduciendo el número de experiencias antiguas debido a que el número de experiencias generadas cada intervalo es de 100 en lugar de 25 como en el escenario de un enlace por celda. No obstante el mecanismo de gestión DER, retiene y reutiliza las experiencias antiguas en el buffer de exploración incrementando la diversidad de los mini-lotes seleccionados resultando en la menor variación del rendimiento comparado con UER, PER y DER.

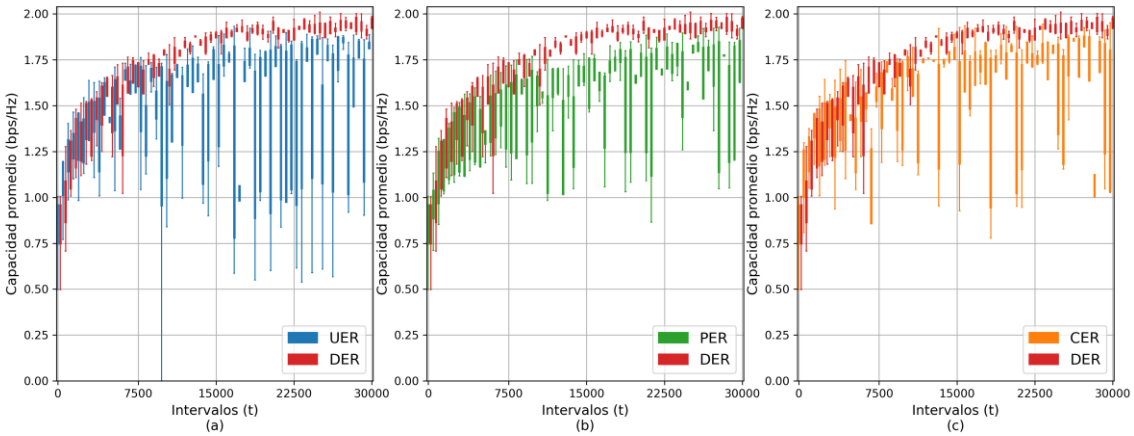

Figura 5.14. Variación de la capacidad promedio de la red durante el entrenamiento del modelo DQN con DER y un buffer de 50K respecto a diferentes mecanismos de gestión de experiencias en un entorno con 25 BS y cuatro enlaces transmisor-receptor en cada celda. (a) UER. (b) PER. (b) CER.

Las funciones de distribución acumulativa (CDF – cumulative distribution function) de la capacidad promedio de los modelos DQN entrenados con diferentes mecanismos de gestión de experiencias (UER, PER, CER y DER) para un entorno de 25 celdas con un enlace en cada celda con diferentes tamaños de buffer (10K y 50K) se muestran en la Figura 5.15. Después del entrenamiento de ajuste se evaluaron los modelos DQN en 500 intervalos de tiempo en la misma condición en la que fueron entrenados. La Figura 5.15(a) muestra que la capacidad alcanzada por todos los modelos entrenados con un buffer de 10K es menor que utilizar un buffer de 50K para el entrenamiento. Por otra parte, la configuración Scratch muestra la mayor capacidad. Este comportamiento se debe a que el modelo se inicializó con los parámetros de la DNN aleatorios y que el parámetro de exploración se inicializa 0.9 y se decrementa de forma lineal durante los primeros 10K episodios hasta mantenerse fijo por el resto del entrenamiento en 0.001. Esta inicialización provoca una mejor exploración que los mecanismos UER, PER, CER y DER con transferencia de parámetros inicial. Sin embargo,

como se mostró en las Figuras 5.9 y 5.10, realizar el entrenamiento con la configuración Scratch provoca una caída de desempeño durante el aprendizaje del modelo. Por otra parte, los modelos entrenados con DER presentan una ganancia de capacidad de la red del 4.22% y 1.25% con respecto al obtenido por UER para un buffer de 10K y 50K, respectivamente.

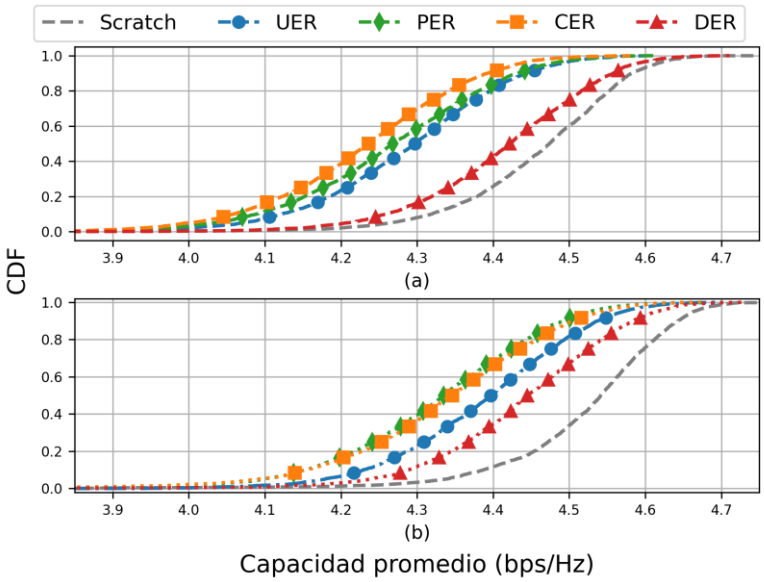

Figura 5.15. Capacidad promedio de la evaluación del modelo DQN con diferentes mecanismos de gestión de experiencias en 500 intervalos de tiempo de la misma condición en la que fueron entrenados en un entorno de 25 celdas con un enlace en cada celda. (a) Buffer de ER tamaño de 10K. (b) Buffer de ER de tamaño de 50K.

La Figura 5.16 muestra la CDF de la capacidad promedio de los modelos DQN entrenados con diferentes mecanismos de gestión para un entorno de 25 celdas con cuatro enlaces en cada celda con diferentes tamaños de buffer. En este caso, se puede observar que el rendimiento del modelo DQN con configuración Scratch presenta el desempeño más bajo, independientemente el tamaño del buffer, debido a que el escenario más interferente considerado durante el entrenamiento de ajuste provoca que no se encuentre la mejor solución. Por otro lado, el entrenamiento con DER mejora el rendimiento de la red en un 24.75% respecto al entrenamiento de ajuste con UER.

La Tablas 5.4 y 5.5 muestran el concentrado de los resultados obtenidos para este experimento considerando los mecanismos de gestión de experiencia (UER, PER, CER y DER) y los tamaños del buffer de ER (10K y 50K). En este caso todos los mecanismos cuentan con el mismo jumpstart (JS) dado que se transfirieron los mismos parámetros para iniciar el entrenamiento de ajuste. Además, el modelo DQN se entrenó solo en una condición

(escenario generado con una semilla aleatoria), mostrando mayor inestabilidad que los resultados de la Tabla 5.3. Para este experimento el mecanismo de gestión de experiencias DER logra el mejor desempeño en todas las métricas de evaluación cuando se reutilizan las experiencias de exploración. Los resultados de las métricas JS, TT, CP, TR, TRTT e IQR

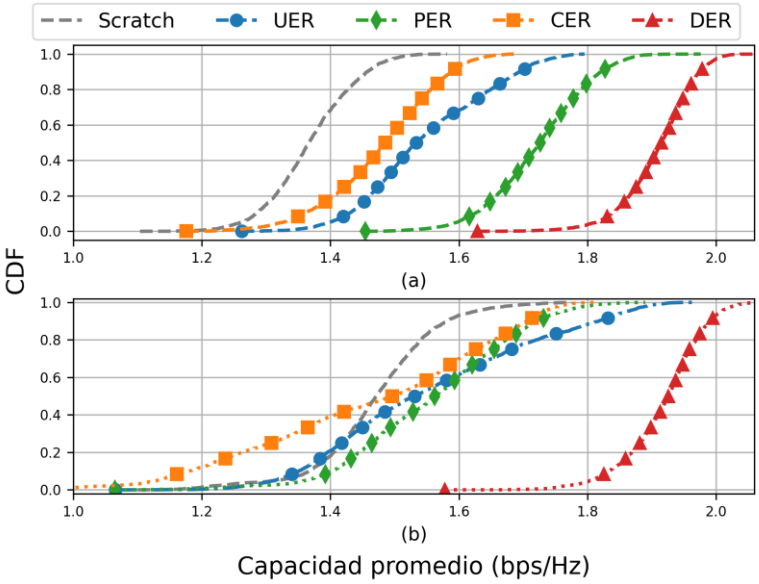

Figura 5.16. Capacidad promedio de la evaluación del modelo DQN con diferentes mecanismos de gestión de experiencias en 500 intervalos de tiempo de la misma condición en la que fueron entrenados en un entorno de 25 celdas con cuatro enlaces en cada celda. (a) Buffer de ER tamaño de 10K. (b) Buffer de ER de tamaño de 50K.

Tabla 5.4. Métricas de desempeño de diferentes esquemas de gestión del entrenamiento del modelo DQN para un entorno con 25 BS y un enlace en cada celda. JS: Jumpstart. TT: Tiempo Transitorio. CP: Capacidad. TR: Tasa de Transferencia. TRTT: Tasa de Transferencia al Tiempo Transitorio. IQR: Rango Inter-cuártil.

	Buffer 50K					
	JS (bps/Hz)	TT (t)	CP (bps/Hz)	TR (%)	TRTT (%)	IQR (bps/Hz)
Scratch	1.7865	17000	3.9083	-	-	0.1319
UER	**2.9067**	2000	4.2911	9.79	55.48	0.0779
CER	**2.9067**	2000	4.2762	9.41	56.37	0.0742
PER	**2.9067**	7000	4.2518	8.78	37.43	0.0807
DER	**2.9067**	**1500**	**4.3461**	**11.20**	**57.11**	**0.0547**
	Buffer 10K					
	JS (bps/Hz)	TT (t)	CP (bps/Hz)	TR (%)	TRTT (%)	IQR (bps/Hz)
Scratch	1.7865	13500	3.9382	-	-	0.2458
UER	**2.9067**	26500	4.1926	6.46	7.91	0.1599
CER	**2.9067**	28500	4.1353	5.00	5.58	0.1616
PER	**2.9067**	25000	4.1539	5.47	7.66	0.1584
DER	**2.9067**	**1000**	**4.3365**	**10.11**	**53.67**	**0.0538**

representan el rendimiento durante el entrenamiento de ajuste. Sin embargo, como se observó en las Figuras 5.9, el rendimiento al final del entrenamiento (intervalo 30K) por otras configuraciones como Scratch puede superar el rendimiento por DER a costa de un menor rendimiento inicial (JS).

Tabla 5.5. Métricas de desempeño de diferentes esquemas de gestión del entrenamiento del modelo DQN para un entorno con 25 BS y cuatro enlaces en cada celda. JS: Jumpstart. TT: Tiempo Transitorio. CP: Capacidad. TR: Tasa de Transferencia. TRTT: Tasa de Transferencia al Tiempo Transitorio. IQR: Rango Inter-cuártil.

	Búfer 50K					
	JS (bps/Hz)	TT (t)	CP (bps/Hz)	TR (%)	TRTT (%)	IQR (bps/Hz)
Scratch	0.4998	30000	1.2349	-	-	0.2541
UER	**0.8003**	30000	1.4533	17.76	17.76	0.3344
CER	**0.8003**	30000	1.5402	24.72	24.72	0.2344
PER	**0.8003**	30000	1.4975	21.26	21.26	0.3538
DER	**0.8003**	**9500**	**1.7354**	**40.52**	**71.78**	**0.088**
	Búfer 10K					
	JS (bps/Hz)	TT (t)	CP (bps/Hz)	TR (%)	TRTT (%)	IQR (bps/Hz)
Scratch	0.4998	30000	1.1108	-	-	0.3047
UER	**0.8003**	30000	1.5538	39.88	39.88	0.2301
CER	**0.8003**	30000	1.4391	29.56	29.56	0.3047
PER	**0.8003**	26500	1.5176	36.62	38.24	0.2874
DER	**0.8003**	**9500**	**1.7048**	**53.47**	**82.25**	**0.1291**

En la Figuras 5.17 a 5.20 se compara la diversidad de las experiencias seleccionadas en el mini-lote durante el entrenamiento del modelo DQN con los mecanismos UER, PER, CER y DER para el entrenamiento de (i) un entorno de 25 celdas con un enlace en cada celda y un buffer de 10K, (ii) 25 celdas con un enlace en cada celda y un buffer de 10K, (iii) 25 celdas con cuatro enlaces en cada celda y un buffer de 10K y, (iv) 25 celdas con cuatro enlaces en cada celda y un buffer de 50K, respectivamente. En las figuras se muestra el mapa de calor para analizar la diversidad del mini-lote de cada configuración. Las variaciones de frecuencia sobre el eje x de cada acción representa la estabilidad de la política de cada configuración (entre menor variación mayor estabilidad). Por ejemplo, para las configuraciones en las que se consideró el tamaño de buffer de 10K (ver Figuras 5.17 y 5.19) se observa un cambio en la magnitud con variaciones entre 50 y 100 de la frecuencias de las acciones las configuraciones UER, PER y CER. En este experimento no se reutilizan experiencias externas, por lo que las experiencias que se seleccionan del mini-lote

corresponden a las experiencias generadas durante el entrenamiento de ajuste. Un entrenamiento estable corresponde al comportamiento observado en la Figura 5.18, en donde las acciones 1, 3 y 10 incrementan su frecuencia y se mantienen estable en el eje x a lo largo del entrenamiento. Estas tres acciones representan las mejores acciones para la condición del entrenamiento de ajuste, por lo que, a pesar del tamaño del buffer de ER limitado, estas acciones se siguen repitiendo y por consecuencia siguen siendo seleccionadas en el mini-lote. Sin embargo, en la Figura 5.18 también se observa que los mecanismos UER, CER y DER no cuentan con información de las acciones 6 a 8 en los mini-lotes. A pesar de que estas

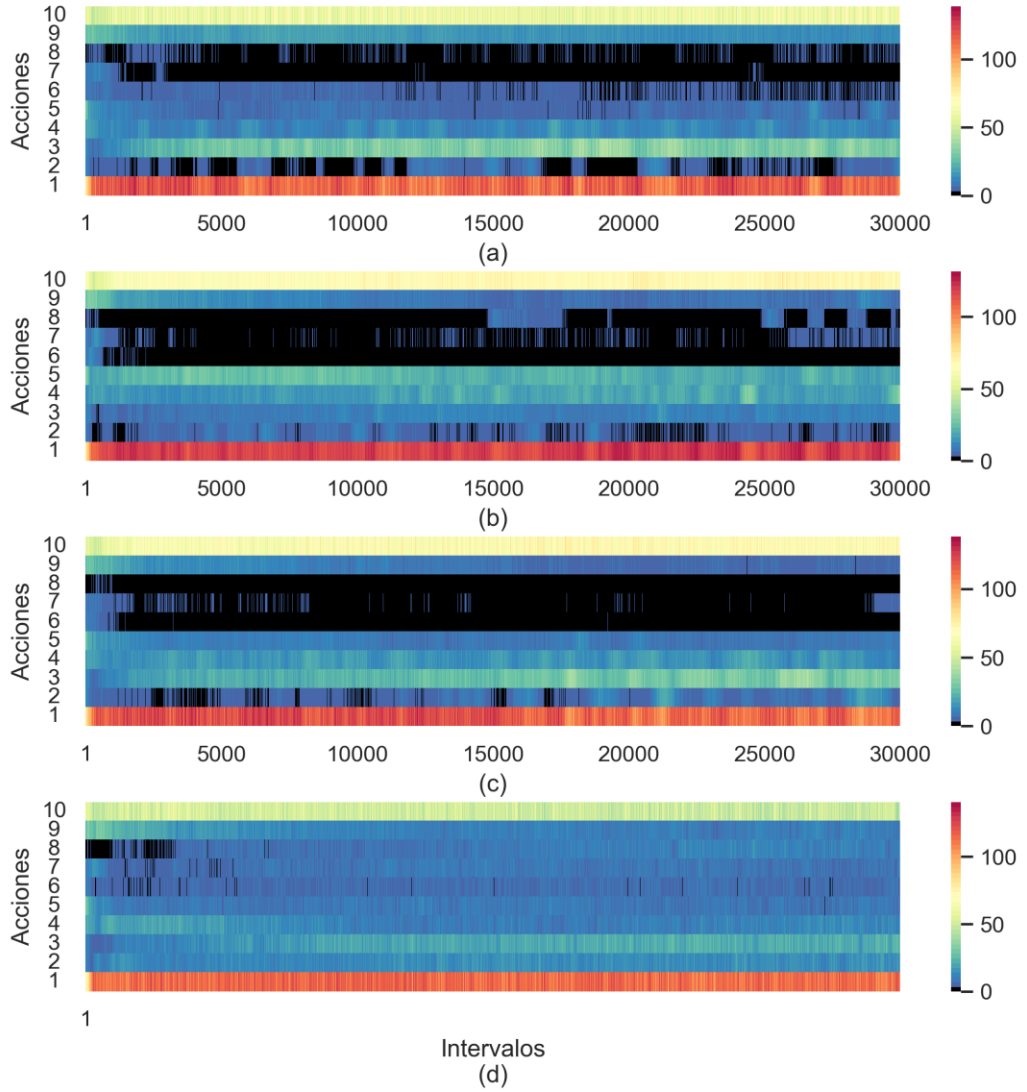

Figura 5.17. Distribución de las acciones seleccionadas en cada mini-lote durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias con un buffer de 10K para un entorno de 25 celdas y un enlace por cada celda. (a) UER. (b) PER. (c) CER. (d) DER.

acciones no son las adecuadas para el entorno de red actual, contar con estas experiencias en el mini-lote permiten ajustar los parámetros con mayor eficiencia, resultando en una reducción de la variación de la capacidad (ver Figura 5.12). Esta falta de diversidad y estabilidad en las frecuencias de las experiencias de los mini-lotes se vuelve más notable cuando se tienen cuatro enlaces por celda (Figuras 4.19 y 4.20). En ese caso, DER logra un

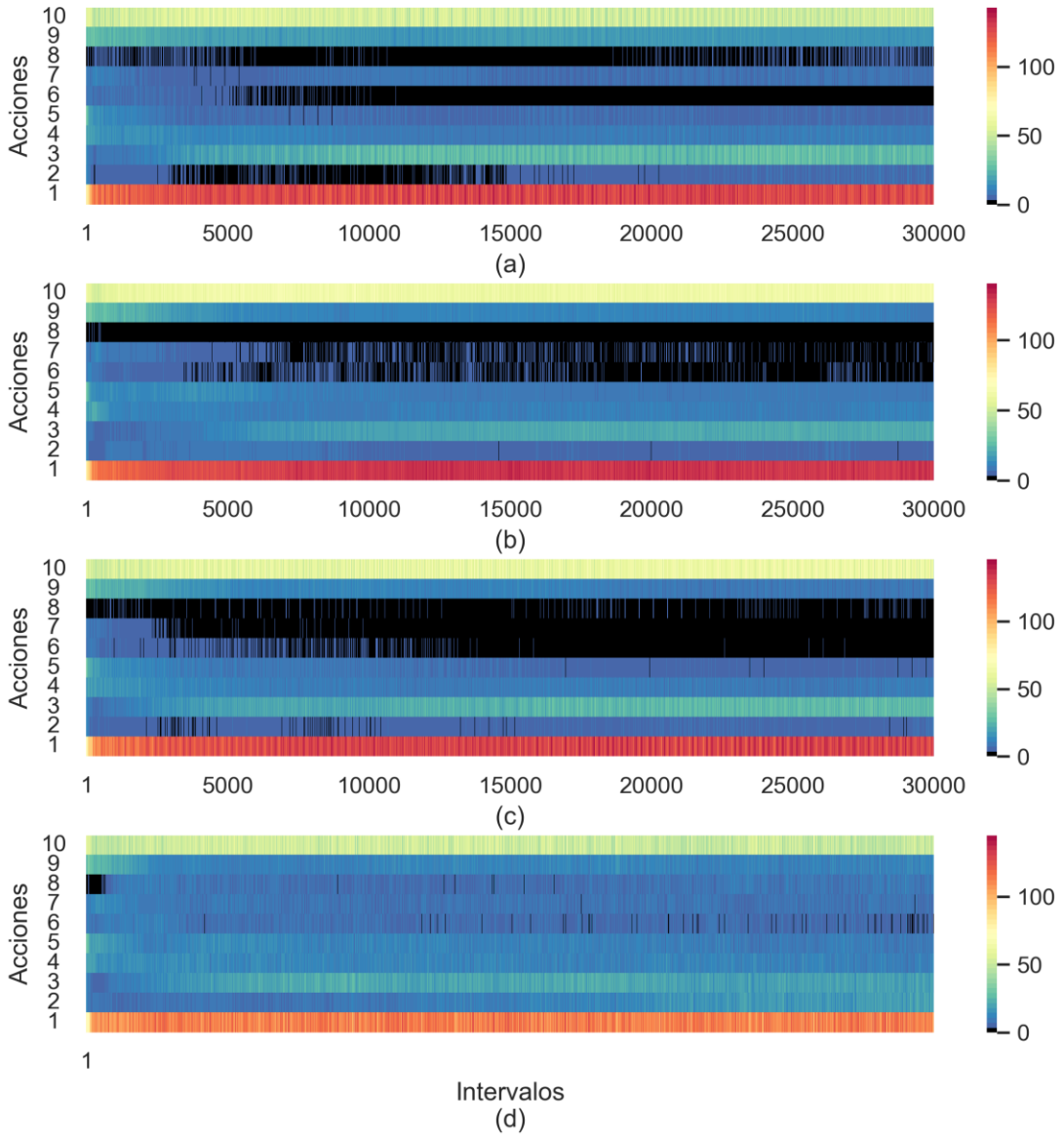

Figura 5.18. Distribución de las acciones seleccionadas en cada mini-lote durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias con un buffer de 50K para un entorno de 25 celdas y un enlace por cada celda. (a) UER. (b) PER. (c) CER. (d) DER.

comportamiento más estable en el eje x a partir del intervalo 17500, mientras que UER, PER y DER no logran estabilizar su política, por ello se observan altas variaciones en las magnitudes de frecuencia para diferentes acciones a lo largo del entrenamiento.

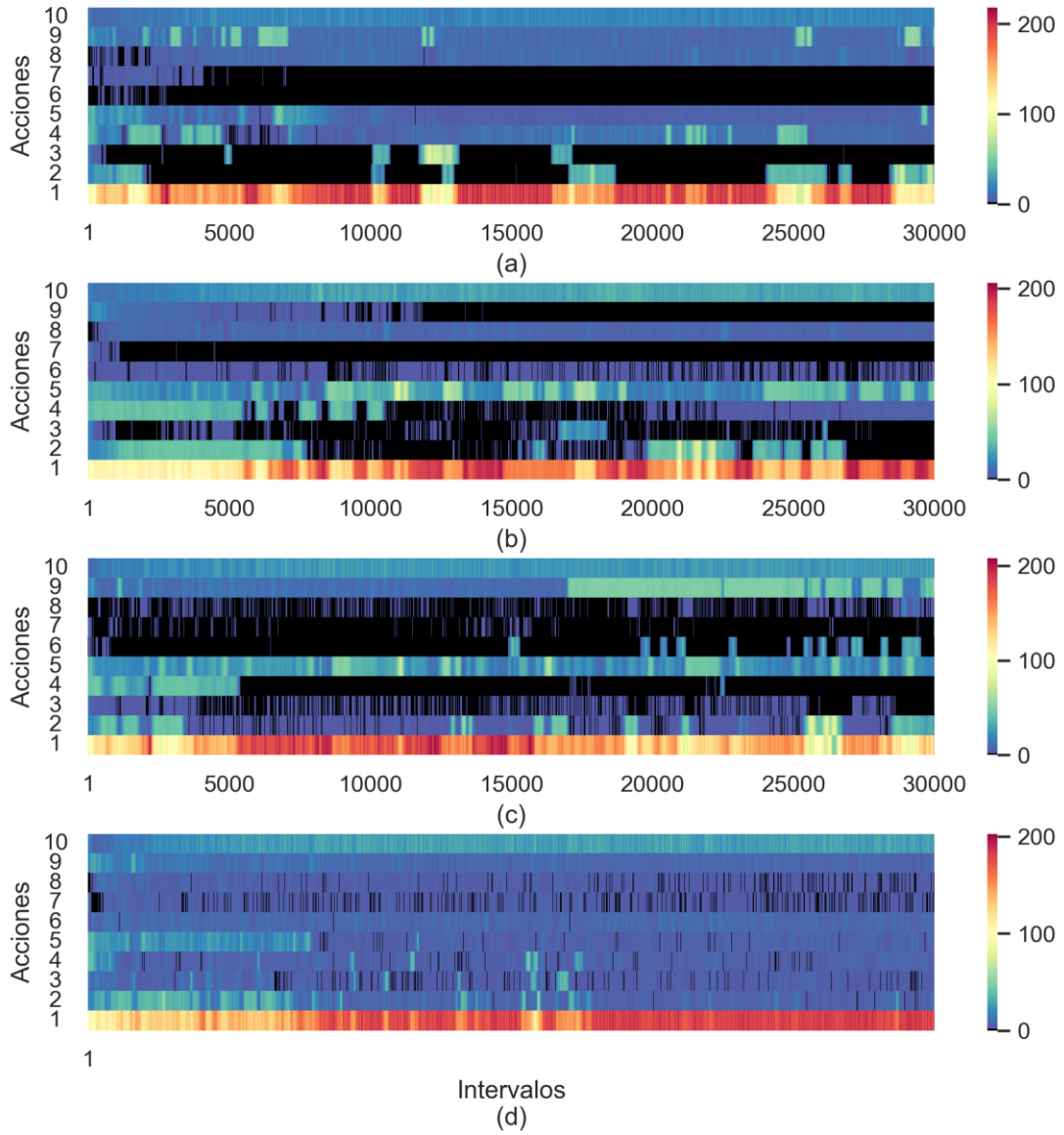

Figura 5.19. Distribución de las acciones seleccionadas en cada mini-lote durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias con un buffer de 10K para un entorno de 25 celdas y cuatro enlaces por cada celda. (a) UER. (b) PER. (c) CER. (d) DER.

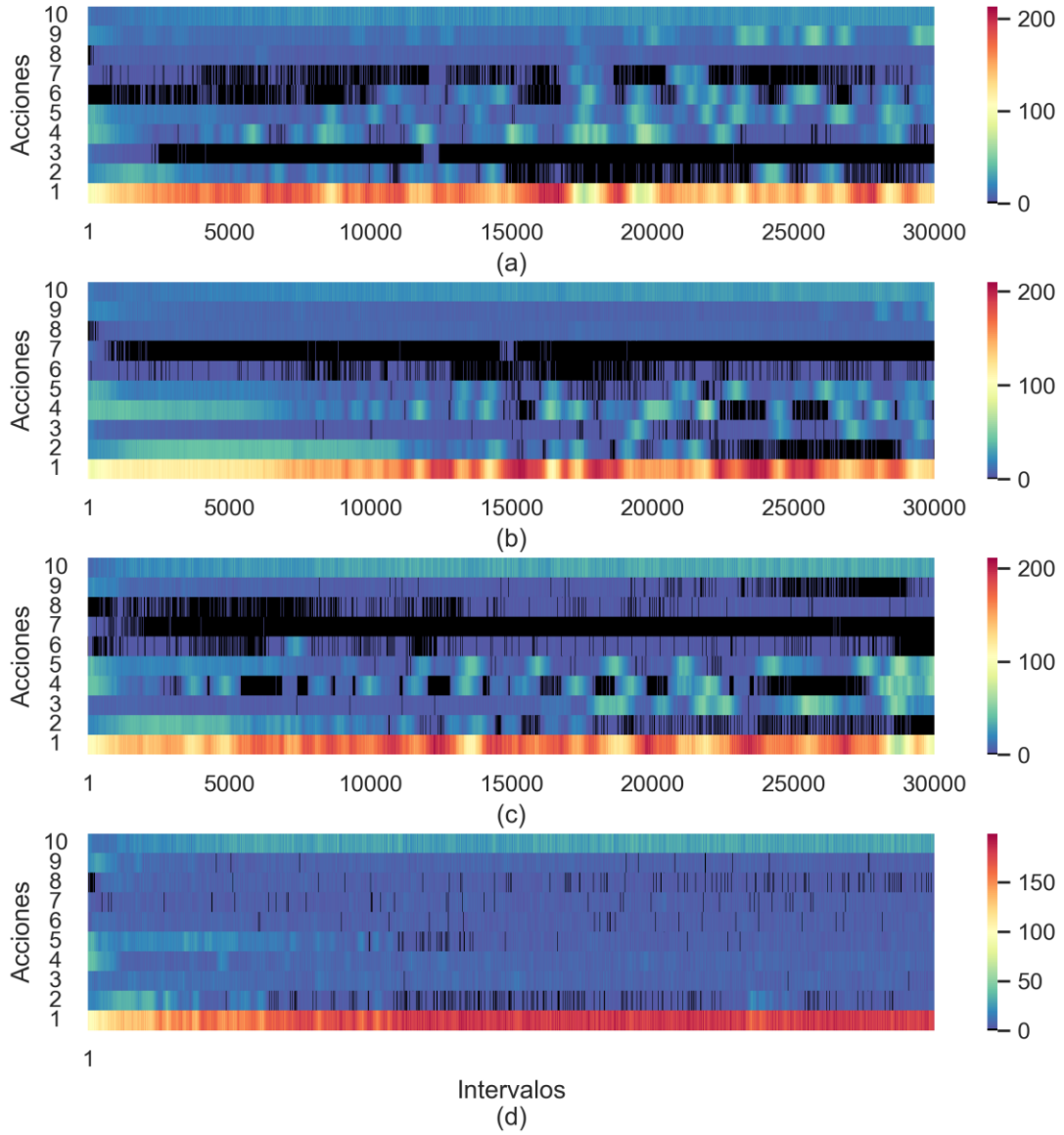

Figura 5.20. Distribución de las acciones seleccionadas en cada mini-lote durante el entrenamiento del modelo DQN con diferentes mecanismos de gestión de experiencias con un buffer de 10K para un entorno de 25 celdas y cuatro enlaces por cada celda. (a) UER. (b) PER. (c) CER. (d) DER.

En la Figura 4.21 se presenta la evaluación de la generalización de las políticas entrenadas con diferentes mecanismos de gestión de experiencias. La evaluación consiste en generar 1000 entornos diferentes con una duración de 10 intervalos. Los resultados muestran una diferencia de capacidad promedio de 2.4 y 2.7 bps/Hz para los escenarios con un enlace por celda y una diferencia de capacidad promedio de 1.3 y 1.0 bps/Hz para los escenarios con cuatro enlaces por celda. En este caso, el mecanismo PER obtiene el mejor rendimiento

para cuatro enlaces por celda mostrando mayor generalización. Este mecanismo es el que tiene un peor desempeño en las evaluaciones del entrenamiento de ajuste. Sin embargo, muestra una mayor robustez al evaluarlo en distintas condiciones de propagación con cuatro enlaces por celda. Por otro lado, el alto desempeño de PER-10K y DER-10K demuestran que el tamaño del buffer de ER puede reducirse sin limitar la robustez de los modelos DQN entrenados.

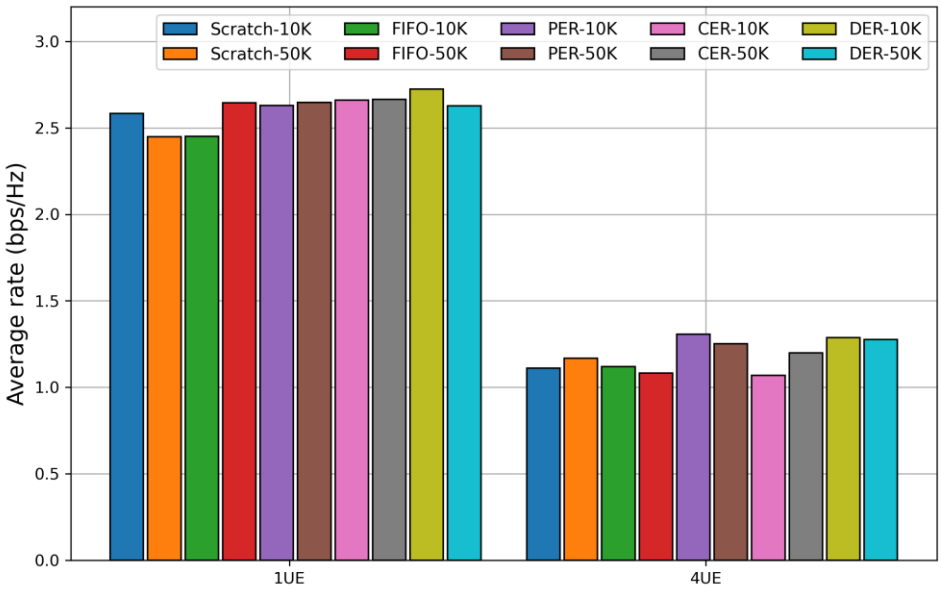

Figura 5.21. Capacidad promedio de los modelos DQN entrenados con diferentes mecanismos de gestión de experiencias evaluados en 1000 condiciones de entorno

5.4.3 Discusión de resultados

Los resultados de este experimento muestran que, a pesar de las diferentes configuraciones, los modelos DQN alcanzan valores de convergencia similar al final del entrenamiento cuando se considera un enlace por celda. No obstante, los mecanismos de gestión de experiencias ayudan a que se logre estabilidad en el rendimiento de la red (menos disminuciones o aumentos abruptos) durante la convergencia del modelo (Figura 5.10). Esta variación del rendimiento se debe a la falta de información del entorno (i.e., experiencias de exploración) y a la selección inadecuada de experiencias de los esquemas UER, PER y CER. Esto es, el mecanismo UER solo mantiene las experiencias más recientes debido a que el tamaño limitado del buffer no le permite mantener experiencias antiguas, mientras que los mecanismos CER y PER priorizan la selección de acciones más recientes limitando la

diversidad del mini-lote. A diferencia de los resultados descritos en la sección 5.2, el modelo DQN transferido se entrena considerando una sola condición, lo que sesga las experiencias generadas y en consecuencia los mecanismos CER y PER empeoran su desempeño de aprendizaje. En contraste, DER retiene, por más tiempo, las experiencias de exploración por lo que la diversidad en el espacio estado-acción en los mini-lotes de experiencias mejoran el aprendizaje, incluso cuando el tamaño del buffer de ER es de poca capacidad.

A pesar de que cuando se incrementa el tamaño del buffer de ER mejora el aprendizaje de todos los mecanismos de gestión de ER, los resultados de rendimiento muestran que retener el mismo número de experiencias y gestionar las experiencias por medio del mecanismo DER logra un mejor entrenamiento de ajuste que el esquema UER. Por lo que, para reducir los requerimientos de memoria de los dispositivos implementando los modelos DQN en redes inalámbricas se requieren mecanismos de ER eficientes.

Por otra parte, con base a los resultados, los mecanismos UER, PER y CER no son adecuados para el entrenamiento de ajuste de los modelos DRL ya que añaden incertidumbre al nivel de calidad de servicio esperado por los proveedores de red. Para evitar la limitante en la inestabilidad de la capacidad de la red de los modelos DRL para las redes inalámbricas y redes futuras, se requieren mecanismos de entrenamiento eficientes para adaptarse a las dinámicas de la red. A pesar de que estrategias como PER podrían implementarse en las radiobases o en los dispositivos celulares para mejorar el aprendizaje de los modelos DRL, estos mecanismos requieren un cómputo adicional sobre el buffer de ER para agregar o priorizar experiencias, llevando a un incremento en el coste computacional que requiere investigarse con mayor detalle. Lo anterior es particularmente un reto para el buffer de ER con baja capacidad o buffer de ER con altas tasas de almacenamiento, tales como los implementados en los esquemas CTDE. Por esta razón, se requiere realizar más investigación y desarrollo de estrategias más simples para acelerar el aprendizaje, reducir la inestabilidad y preservar la diversidad del espacio estado-acción en redes B5G.

5.5 Experimento 4: Evaluación de la Transferencia de instancias para maximizar la eficiencia energética de la red

En este experimento se evalúa el tiempo transitorio al reutilizar experiencias de modelos entrenados en diferentes entornos de red con el fin de validar la hipótesis $H1_0$. Las

experiencias reutilizadas consisten en experiencias expertas que se utilizan durante el entrenamiento del modelo DQN para resolver el problema de asignación de potencia en una red móvil B5G con el propósito de maximizar la eficiencia energética (EE).

5.5.1 Configuración del escenario

El entrenamiento del modelo DQN consiste en 5K episodios con una duración de 10 intervalos cada episodio. La política del modelo DQN se evalúa cada cinco episodios en un conjunto de validación de 100 intervalos de tiempo. Las potencias máximas y mínimas se establecen de 23 dBm y -20 dBm respectivamente, con celdas de radio de cobertura de 30 metros. El tamaño del mini-lote de experiencias es de 500. El resto de los parámetros permanecen iguales a los definidos en la Sección 5.1.

A diferencia de la sección anterior, en este experimento solo se evalúa la transferencia de instancias. Con base a la nomenclatura definida para los esquemas de transferencias, NIT corresponde a la curva de referencia o al entrenamiento desde cero. Para propósitos de comparación se implementó una variante de EIT. En lugar de inicializar el entrenamiento con el buffer experto, las experiencias expertas se almacenan gradualmente en conjunto con las experiencias de los agentes que interactúan con el entorno. Por ejemplo, para el caso de 25 agentes, en cada intervalo de tiempo se almacenan 50 experiencias en el buffer de ER de las cuales 25 experiencias pertenecen a los agentes que interactúan con el entorno y 25 experiencias corresponden al buffer experto. Las variantes de EIT se indican por números arábigos del 1 al 4 y corresponden a experiencias expertas de modelos DQN entrenados en escenarios con 4, 9, 16 y 25 celdas con un enlace en cada celda, denominados EIT-1, EIT-2, EIT-3 y EIT-4.

5.5.2 Resultados del experimento

La Figura 5.22 muestra la EE que logra el modelo DQN bajo el esquema EIT con diferentes tamaños de buffer para un escenario de 25 celdas con un enlace por cada celda. Los resultados muestran que el punto de convergencia es similar para todas las variantes del mecanismo EIT sin importar la configuración del buffer de ER. Sin embargo, agregar experiencias de otros modelos con las variantes de EIT incrementa el rendimiento de la red en los primeros 6200 intervalos comparados con NIT. No obstante, con EIT-1 se deteriora el

rendimiento a partir del intervalo 6200, alcanzando un rendimiento menor que el obtenido por NIT. Por otra parte, la configuración EIT-3 muestra el mejor rendimiento con una convergencia a partir del intervalo 15400 y una mejora del rendimiento de hasta 117% en los primeros 1500 intervalos respecto a NIT.

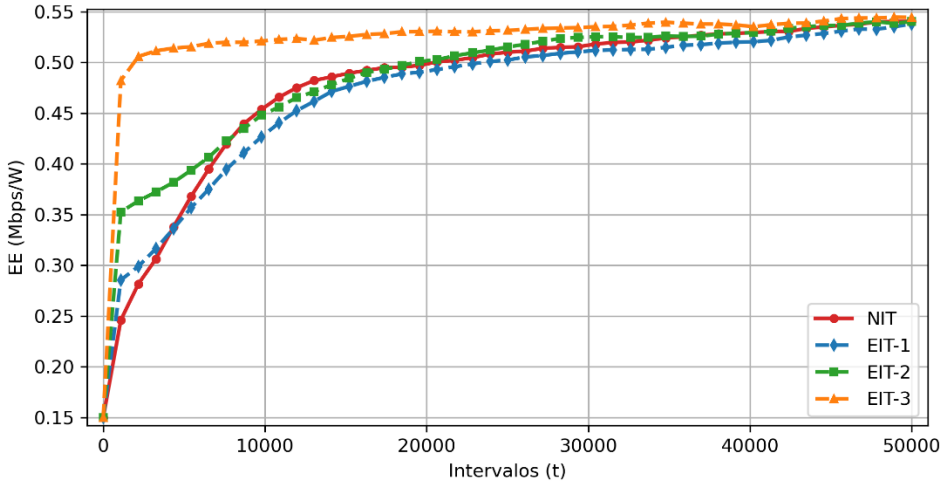

Figura 5.22. Rendimiento de la evaluación del modelo DQN bajo el esquema EIT con diferentes buffer para un escenario con 25 celdas y un enlace por celda.

Para la Figura 5.23 se consideró un escenario de 36 celdas con un enlace por celda. Similar al experimento anterior la reutilización de las experiencias de entornos con menor densidad de celdas (i.e., EIT-1 y EIT-2) muestran una mejora inicial seguido por un desempeño inferior a NIT a partir del intervalo 4K. Lo anterior se debe al sesgo ocasionado por reutilizar experiencias de modelos de escenarios más simples provocando una asignación de potencia inadecuada para entornos de red más interferentes. A pesar de este rendimiento obtenido por el modelo DQN con EIT-1 y EIT-2, el comportamiento de la curva muestra valores similares de convergencia que NIT mostrando la adaptación del modelo DQN. Por otro lado, las configuraciones EIT-3 y EIT-4 muestran una aceleración del aprendizaje de alrededor de 36.23% y 46.95% en los primeros 5K episodios, respectivamente.

En la Figura 5.24 se presenta el efecto sobre el entrenamiento del modelo DQN al utilizar diferentes niveles de potencia para un escenario con 25 celdas y un enlace en cada celda. La reducción de niveles de potencia reduce el espacio de acción. Los resultados muestran que implementar EIT-3 mejora de igual medida el rendimiento de la red independientemente del número de niveles de potencia del modelo DQN. Por otra parte, el

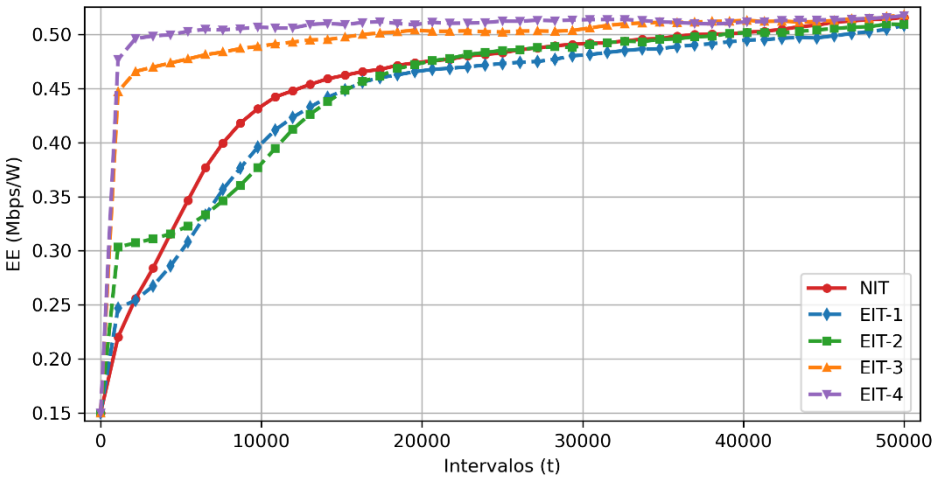

Figura 5.23. Rendimiento de la evaluación del modelo DQN bajo el esquema EIT con diferentes buffer para un escenario con 36 celdas y un enlace por celda.

modelo NIT muestra que el rendimiento inicial durante los primeros 21K intervalos mejora un 7.4% y 13.61% al reducir los niveles de potencia a 5, mientras que el rendimiento se reduce un 5.47% al incrementar los niveles de potencia a 15 con respecto a contar con 10 niveles de potencia.

Por otra parte, el modelo DQN con NIT muestra que el rendimiento inicial durante los primeros 21K intervalos mejora un 7.4% y 13.91% al reducir los niveles de potencia a 5 con respecto a NIT con 10 y 15 niveles de potencia, respectivamente. Como se observó en los mapas de calor de la sección 5.4, cada condición muestra un perfil de potencias diferentes en el cual se eligen pocas acciones con mayor frecuencia. A pesar de que los resultados muestran una mejora cuando se reduce la cantidad de niveles de potencia es necesario medir el efecto de esta reducción en diferentes condiciones de red para determinar el número de parámetros de salida óptimo para mejorar el aprendizaje de los modelos DQN. Sin embargo, el número de parámetros de salida no afecta el aprendizaje al reutilizar las experiencias expertas. Lo anterior se debe a que las experiencias expertas cuentan con el conocimiento de las mejores acciones, lo que permite ajustar la política del modelo DQN rápidamente para reforzar las acciones más adecuadas y disminuir la frecuencia de las peores acciones independientemente del número de acciones totales en el espacio de acción.

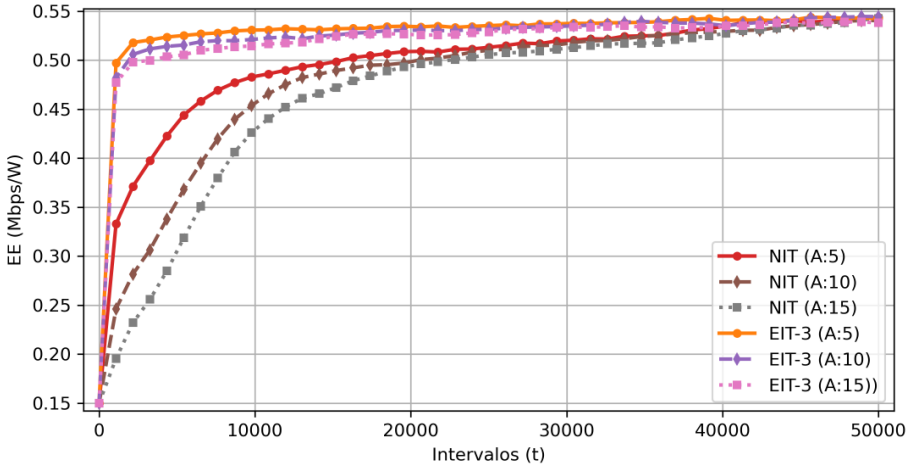

Figura 5.24. Rendimiento de la evaluación del modelo DQN con diferentes niveles de potencia un escenario con 25 celdas y un enlace por celda.

5.5.3 Discusión de resultados

La conclusión de este experimento establece que al reutilizar las experiencias expertas se logra una aceleración de la convergencia del modelo DQN cuando se utiliza el esquema EIT en comparación con el esquema EIT. Cabe señalar que en este caso no se realizó una transferencia de parámetros, por lo que los resultados de NIT no cuentan con el sesgo de un modelo previamente entrenado. De igual forma, el modelo DQN con EIT se entrenó con un modelo inicializado al azar. Lo anterior muestra una aceleración de convergencia en pocos intervalos al reutilizar las experiencias más similares (EIT-3 o EIT-4). Además, el valor de épsilon se reinicia, generando experiencias de exploración (diversidad en el espacio de acción) desde el inicio del entrenamiento. Por otra parte, el conocimiento de las experiencias menos similares muestra una mejora parcial en el rendimiento de la red al inicio del entrenamiento. Lo anterior muestra el potencial en el diseño y reutilización de experiencias sintéticas o gestión de las experiencias generadas en la red para acelerar el proceso de aprendizaje al ajustar nuevos modelos DQN.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Resumen

La evolución de las redes móviles celulares se encamina al paradigma de la comunicación inteligente con automatización y control de tareas en tiempo real de forma confiable y sostenible. Estas futuras generaciones de telecomunicación alimentadas por las técnicas de IA brindarán un servicio global con mayores velocidades de datos, menor retardo y conectividad total para los distintos dispositivos en la red. Para lograr la visión de red inteligente mencionada, es necesario incrustar las técnicas de IA en el diseño de los sistemas futuros para automatizar las funciones de red tales como la asignación óptima de recursos, la predicción del comportamiento de los suscriptores móviles o análisis de los patrones de tráfico, la detección de fallas, entre otros. Los beneficios de la automatización de las funciones de las redes B5G también incrementará la cantidad de decisiones incorrectas tomadas por las técnicas de IA. Para minimizar las posibles fallas en las funciones de red es necesario considerar los factores como lo son el tipo de entidades, la densidad de las redes, los requerimientos de servicio o las tecnologías de comunicación implementadas. Además, es necesario analizar y monitorear el comportamiento de la red B5G controladas por las técnicas inteligentes. Sin embargo, la mayoría de los trabajos de investigación que implementan modelos inteligentes para realizar la toma de decisiones logran resultados prometedores comparados con los algoritmos convencionales para resolver un problema de un entorno de red específico. La consideración de múltiples factores en el entorno de red o

entornos con diferentes tipos de redes (e.g., redes celulares y redes vehiculares) incrementa el número de variables que los modelos de IA requieren procesar para obtener la mejor solución al problema atendido. Este incremento de variables requiere modelos de IA más complejos que requieren un coste computacional mayor y un tiempo prolongado para ajustar sus parámetros denominado como entrenamiento del modelo. A pesar de que es posible adaptar paulatinamente los modelos de IA con la información generada en la red para entrenar estos modelos complejos, las redes B5G requieren una toma de decisiones rápidas. Es decir, modelos de IA de menor complejidad que se adapten rápidamente a los cambios de los sistemas B5G como lo son el incremento en el número y tipo de dispositivos conectados, el despliegue de nuevas estaciones base, los requerimientos dinámicos de calidad de servicio y a las fluctuaciones del tráfico en la red.

Las técnicas de IA son apropiadas para resolver el problema de asignación de recursos. La asignación de recursos es un problema estudiado desde los sistemas móviles de segunda generación (2G), en el que los nodos deciden como distribuir los recursos radioeléctricos a sus usuarios atendidos para mejorar la capacidad de la red. Sin embargo, este problema se vuelve más complejo a medida que más transmisores reutilizan la misma banda espectral para transmitir información, lo cual se vuelve inevitable a medida que se incrementa la densificación de usuarios y nodos en la red. Por lo que encontrar soluciones óptimas es un proceso computacionalmente complejo, los algoritmos de optimización aumentan su complejidad debido a las condiciones dinámicas de la red, el número de usuarios, nodos o requerimientos de calidad de servicio. Este compromiso entre rapidez en la toma de decisiones y adaptabilidad a las dinámicas del entorno abre el camino al uso de técnicas de aprendizaje automático (ML) en redes B5G. La rapidez en la toma de decisiones es una de las principales ventajas respecto a los algoritmos de optimización iterativos implementados para la asignación de recursos.

Como se describió en el capítulo dos, las técnicas de inteligencia artificial, en particular de ML, aplicadas para resolver el problema de asignación de recursos se enfocan principalmente en la extracción de la información de la red y la toma de decisiones. Implementar estás técnicas de ML en producción requieren un proceso de entrenamiento, en el cual ajustan sus parámetros internos para optimizar su rendimiento en la toma de decisiones. Los modelos de ML para la asignación de recursos se clasifican en tres grupos:

técnicas basadas en redes neuronales artificiales (ANN), técnicas basadas en aprendizaje reforzado (RL) y técnicas basadas en aprendizaje por refuerzo profundo (DRL). Las técnicas basadas en ANN se entrenan con una base de datos óptima. La base de datos se obtiene a partir de resolver el problema de asignación de recursos cientos de veces por medio de un algoritmo de optimización. Después, los algoritmos basados en ANN aprenden a imitar el comportamiento de estos algoritmos de optimización resultando en un tiempo de ejecución menor que estos. Este requerimiento de contar previamente con una base de datos es uno de los desafíos principales de los modelos basados en ANN, los cuales no son posibles de generar en tiempo real debido a los cambios dinámicos de los entornos B5G y al tiempo que requieren los algoritmos de optimización en encontrar iterativamente la solución óptima. Por otra parte, los modelos basados en RL generan su propia base de datos por medio de la interacción directa con el entorno de red. Este proceso de interacción a prueba y error genera conjuntos de datos con mejores soluciones a medida que el proceso de entrenamiento continúa permanentemente. Sin embargo, a medida que la red se vuelve más compleja, estos modelos se vuelven difíciles de entrenar porque el espacio de búsqueda de la mejor solución se incrementa exponencialmente. Por lo anteriormente expuesto, las técnicas basadas en DRL son los modelos de ML que mayormente se han estudiado para evaluar su eficacia en la resolución del problema de asignación de recursos en redes B5G. Los modelos basados en DRL logran un mejor rendimiento que los modelos basados en RL en entornos de red más complejos y logran adaptarse a los cambios de los entornos B5G a diferencia de los modelos basados en ANN. A pesar de que los modelos basados en DRL logran ajustar sus parámetros para adaptarse a las condiciones del entorno en que son implementados, las decisiones incorrectas deben ser minimizadas para garantizar el nivel de calidad de servicio requerido por los operadores móviles. En los sistemas B5G, los modelos basados en DRL requieren ejecutar entrenamientos de ajuste para adaptarse en el menor tiempo posible y de manera confiable a las fluctuaciones de los sistemas inalámbricos. Es decir, el entrenamiento de ajuste debe alcanzar el mejor rendimiento y disminuir la cantidad de decisiones incorrectas que degraden el rendimiento de la red durante el proceso de aprendizaje.

Por lo anterior, en este trabajo de tesis se propuso un método para acelerar y estabilizar la etapa de aprendizaje del modelo DQN (modelo basado en DRL) para la asignación de potencia en una red móvil celular. Para considerar las fluctuaciones de red se

diseñó un protocolo de evaluación que genera diferentes condiciones de red cada cierto intervalo. Estos cambios de red provocan que el modelo DQN requiera entrenamientos de ajuste para adaptar sus parámetros y mejorar el rendimiento de la red al asignar la potencia óptima. El protocolo de evaluación permite analizar los efectos en el rendimiento de la red durante el entrenamiento de ajuste al controlar los cambios de la red de manera que es posible evaluar y comparar los tiempos transitorios y la variación en la capacidad de la red de diferentes modelos DQN. Durante el entrenamiento, se generan experiencias por medio de la interacción a prueba y error entre la entidad que ejecuta el modelo DQN y el entorno B5G. En esta interacción, la entidad toma decisiones aleatorias (experiencias de exploración) para explorar el espacio de búsqueda y encontrar mejores soluciones que permitan ajustar el modelo durante el entrenamiento. En caso contrario, cuando la entidad toma la decisión con base al conocimiento del modelo, las experiencias generadas por la interacción con el entorno B5G se denominan experiencias de explotación. Dado que las experiencias de exploración se generan a partir de la toma de decisión de forma aleatoria, tienden a degradar y causar variaciones el rendimiento de la red. Sin embargo, las experiencias resultantes de la exploración de los modelos DQN contienen información relevante que permite ajustar sus parámetros para evitar las acciones que degraden el rendimiento de la red y reforzar las acciones que propicien un alto rendimiento de la red en la toma de decisiones futuras. Esta estrategia de generación de experiencias de exploración y explotación se vuelve inviable en entornos reales por la incertidumbre en el rendimiento de la red causada por generación de experiencias de exploración. Por tal motivo, los modelos DQN se entrenan en entornos de simulación en una fase inicial y después se transfieren a otros entornos como: sim2sim, para la transferencia a un entorno de simulación o sim2real, para la transferencia a un entorno real. Sin embargo, para que el modelo DQN ajuste sus parámetros en el entorno en que fue transferido, se requiere de un entrenamiento de ajuste en el cual se generen experiencias de exploración. Durante el entrenamiento de ajuste, se limita la cantidad de experiencias de exploración para reducir el efecto negativo en el rendimiento causado por la aleatoriedad de las experiencias de exploración y su vez permitir que el modelo genere experiencias que le permitan optimizar su rendimiento en el nuevo entorno. Esta limitación de las experiencias de exploración provoca que las experiencias del buffer de ER gestionado por el mecanismo de gestión convencional (i.e., mecanismo FIFO) retenga poca información de las

experiencias de exploración. Por tal motivo, para mejorar la gestión del buffer de ER en el modelo DQN y reutilizar las experiencias de exploración, se propuso un buffer de repetición de experiencias dual (DER). Con el buffer DER, las experiencias de exploración se almacenan en un buffer independiente y se reutilizan con mayor frecuencia, reforzando el conocimiento obtenido por las experiencias de exploración. Además, con el mecanismo DER, se reemplazan con mayor frecuencia las experiencias almacenadas en el buffer de explotación, generando nueva información que describen mejor el entorno actual.

La propuesta DER se evaluó en 3 experimentos considerando diferentes cambios en las condiciones de red de forma controlada siguiendo el protocolo de evaluación diseñado en el capítulo 4. Además del mecanismo DER, se evaluó el efecto de reutilizar experiencias generadas para resolver el problema de asignación de recursos en otros entornos de red, denominadas como experiencias expertas. El esquema de transferencias expertas (EIT) almacena las experiencias expertas en el buffer de ER en conjunto con las experiencias generadas por los modelos DQN. Además del esquema EIT se añadió una variante denominada repetición de experiencias dual filtradas (FDER) para reutilizar las experiencias expertas en conjunto con el mecanismo DER. La reutilización de experiencias mostró una reducción del tiempo transitorio (es decir, el tiempo requerido para ajustar el modelo a las nuevas condiciones de la red) y en la variación de la capacidad. Sin embargo, para implementar EIT y FDER se requiere reutilizar experiencias expertas, lo cual limita su implementación al requerir estas experiencias previamente, similar a los modelos basados en ANN. Sin embargo, estos beneficios muestran oportunidades en los trabajos de investigación relacionados con la implementación y diseño de técnicas que generen experiencias sintéticas que podrían simular el buffer experto. Por último, se evaluó la eficiencia energética (EE) con un modelo DQN utilizando el esquema EIT. Para las experiencias expertas se entrenaron modelos en entornos de red con densidades menores al entorno de implementación con el fin de evaluar su efecto en el aprendizaje del modelo. Los resultados mostraron que cuando se reutilizan experiencias se mejora el entrenamiento con EIT con densidades de estación base similar, mientras que el rendimiento al reutilizar experiencias con densidades muy diferentes entre sí afecta el rendimiento del modelo. Por lo que, para mejorar el aprendizaje al reutilizar experiencias es necesario analizarlas e implementar estrategias eficientes que permiten decidir cuales experiencias reutilizar en cada nueva condición que se presente.

6.2 Hipótesis presentadas

En esta Tesis se realizaron diferentes experimentos para validar las siguientes hipótesis:

- Hipótesis nula 1 ($H1_0$): La combinación de un mecanismo de gestión de experiencias y la reutilización de experiencias adquiridas bajo diversas condiciones (ubicación de UE y CSI) no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia en redes móviles B5G.
- Hipótesis nula 2 ($H2_0$): La implementación de un mecanismo de gestión de experiencias que retenga las experiencias en dos buffers independientes (de exploración y de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia.
- Hipótesis nula 3 ($H3_0$): La implementación de un mecanismo de gestión de experiencias que retenga experiencias en dos buffers independientes (i.e., buffer de exploración y buffer de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá la variación de la capacidad de la red durante la asignación de potencia en los modelos Deep Q-Network.

Con base en los resultados obtenidos en los experimentos de la sección 5, las hipótesis $H1_0$, $H2_0$ y $H3_0$ quedan rechazadas parcialmente debido a que es necesario una investigación más exhaustiva que consideré un mayor número de entornos de red. En general, los resultados muestran que el mecanismo de gestión con dos buffers y la reutilización de experiencias expertas reducen el tiempo transitorio de los modelos Deep Q-Network durante el aprendizaje de nuevos entornos ($H1_0$ y $H2_0$). Mientras que la variación de la capacidad de la red es reducida al implementar un mecanismo de gestión de experiencias con dos buffers independientes ($H3_0$).

6.3 Conclusión

En general, implementar un buffer de ER dual en el modelo DQN reduce hasta un 50% el tiempo transitorio respecto implementar el buffer de ER convencional. Es decir, el modelo DQN alcanza el rendimiento umbral utilizando menor información y con un menor coste computacional. Además, la variación del rendimiento se reduce hasta en un 25%, lo

que muestra una mayor confiabilidad al reducir los decrementos abruptos del desempeño de la red al implementar el buffer de ER dual. Por otra parte, los resultados obtenidos también muestran los beneficios de la reutilización de experiencias generadas en entornos similares y los beneficios de mantener la diversidad del buffer de ER. Con base a estos resultados, se establece que existe un compromiso entre la reutilización de experiencias de entornos similares y el rendimiento de la red, lo que permite acelerar el aprendizaje de los modelos DQN y mejorar el rendimiento de la red. Por lo tanto, con base a los resultados obtenidos en esta tesis se concluye lo siguiente:

- Para ajustar los modelos DQN con el fin de optimizar el uso de los recursos radioeléctricos para adaptarse a los cambios del entorno de red, es recomendable segmentar las experiencias del buffer de ER. Esta segmentación del buffer de ER permite preservar el conocimiento de diferentes entornos de red celular y a su vez ajustar la política del modelo considerando el conocimiento experto obtenido en etapas de entrenamiento anteriores. Además, reutilizar estas experiencias segmentadas en conjunto con las experiencias que genera el modelo DQN al interactuar con el entorno de implementación reduce el tiempo transitorio y reduce la variación de la capacidad de la red durante el aprendizaje de nuevos entornos.
- Una segmentación selectiva reduce el requerimiento del tamaño del buffer sin reducir el rendimiento de los modelos DQN para la asignación de recursos. La reducción del tamaño del buffer permite la implementación de los modelos DQN en entidades con menor capacidad computacional, como dispositivos celulares o dispositivos IoT. Además, con una segmentación selectiva como el mecanismo DER o el mecanismo FDER se incrementa la diversidad de las experiencias reduciendo la variación de la capacidad y reduciendo el tiempo transitorio durante el entrenamiento de ajuste para adaptarse a los cambios de la red.

6.3.1 Pregunta de investigación

Dados los cambios constantes del entorno en los sistemas B5G, las decisiones del modelo DQN durante la asignación de recursos deben coincidir con las condiciones del canal para eficientizar el uso de los recursos radioeléctricos en el proceso de asignación de recursos

de las redes celulares. Para tomar la mejor decisión de asignación de recursos, los modelos DQN requieren ajustar sus políticas con base al conocimiento adquirido y almacenado en el buffer de experiencias. Sin embargo, dado los cambios del entorno constantes de los sistemas B5G, los mecanismos de gestión de experiencias convencionales retienen las experiencias más recientes desechando experiencias relevantes en lugar de reutilizarlas para mejorar el aprendizaje. Por tal motivo, en esta tesis de propuso gestionar las experiencias del modelo DQN con un buffer DER para la asignación de potencia en redes celulares con el fin de resolver la siguiente pregunta de investigación:

- ¿De qué manera influye el mecanismo de gestión de experiencias basado en doble buffer en el tiempo transitorio y en la estabilidad de la capacidad de la red durante el aprendizaje de nuevas políticas de asignación de potencia en redes celulares?

Con base a los resultados de experimentación se concluye que el mecanismo de doble buffer o DER permite seleccionar experiencias más diversas respecto al espacio de acción, ajustando los parámetros del modelo DQN al considerar un mini-lote más diverso. Dada la reutilización constante de las experiencias de exploración, el ajuste de los parámetros considera constantemente estas experiencias reduciendo la variación entre los intervalos de actualización de los parámetros del modelo. Esto a su vez reduce la variación de la capacidad resultando en un entrenamiento de ajuste más estable. Como consecuencia de la estabilidad del aprendizaje, el modelo DQN alcanza un punto de convergencia en un menor tiempo transitorio.

6.4 Limitantes

A partir de los resultados obtenidos se detectaron las siguientes limitantes:

- En los experimentos no se consideró un retardo en la generación de experiencias, ni las posibles fallas en la información enviada. Es decir, en un cada intervalo todos los agentes (entidades ejecutando el modelo DQN) interactúan con el entorno y envían sus experiencias al buffer de ER localizado en el controlador central. La consideración del retardo en él envió/generación de experiencias tendría un efecto en la convergencia del modelo lo cual podría causar variaciones

en los resultados o requerir un entrenamiento con un mayor número de intervalos para llegar a la convergencia.

- En los experimentos se consideraron diferentes semillas para generar los escenarios del entrenamiento de ajuste. Sin embargo, no se realizó una evaluación con diferentes semillas para evaluar los efectos de transferencia bajo diferentes condiciones. Por lo que las mejoras obtenidas al implementar el mecanismo DER pueden variar en magnitud. Sin embargo, con base a los mecanismos de referencia, se observa una mejora en general del mecanismo DER para los distintos experimentos por lo que el comportamiento de las curvas no se ve afectado independientemente de la condición considerada para el escenario de evaluación.

6.5 Trabajo futuro

A partir de los resultados obtenidos se detectaron los siguientes trabajos futuros:

- Los resultados mostraron una mejora del aprendizaje respecto a la aceleración de la convergencia en los distintos escenarios de evaluación al considerar la diversidad en el espacio de acción y la implementación del buffer filtrado. Sin embargo, estas experiencias se generaron a partir de las condiciones de evaluación o de entornos previamente entrenados. Debido al alcance del proyecto, no se analizó el efecto de la generación de experiencias sintéticas para incrementar la diversidad del espacio de acción en el buffer de ER. El incremento de datos por medio de estrategia de generación de experiencias sintéticas, tales como GANs, podría incrementar la diversidad del espacio de acción acelerando el entrenamiento y a su vez evitando los mínimos locales al generar nuevas experiencias. Por lo que en trabajos futuros se considera implementar técnicas de aumentación de datos para mejorar la etapa aprendizaje de los modelos DQN en entornos sin experiencias expertas.
- A pesar de las mejoras en el rendimiento de la red por reutilizar experiencias expertas. No se analizaron los efectos en el aprendizaje ni la aportación en el rendimiento de la red de cada una de las experiencias seleccionadas en el mini-lote del buffer de ER. Por ejemplo, el mecanismo DER muestra una aceleración

de la convergencia mientras que el mecanismo PER muestra un mejor resultado en las pruebas de generalización. Por lo que analizar el compromiso entre la generalización de las políticas aprendidas y la plasticidad del modelo DQN con base a la selección de experiencias podría servir para diseñar un mecanismo de selección más adecuado para mejorar la adaptación de nuevos entornos. Es decir, seleccionar ciertas experiencias segmentadas con base a las características de las experiencias que se generaron durante la interacción con el entorno podría reducir el tiempo transitorio de los modelos DRL o añadir robustez en entornos dinámicos.

6.6 Productos Académicos

A partir del trabajo en esta tesis se derivaron los siguientes productos académicos.

Artículos científicos:

- Andrade, Á. G. and Anzaldo, A. Accelerated resource allocation based on experience retention for B5G networks. J. Netw. Comput. Appl. 213. 103593. 2023. https://doi.org/10.1016/j.jnca.2023.103593
- Anzaldo, A. and Andrade, Á. G. Experience Replay-based Power Control for sum-rate maximization in Multi-Cell Networks. IEEE Wirel. Commun. Lett. 11. 11. 2350-2354. 2022. https://doi.org/10.1109/LWC.2022.3202904
- Anzaldo, A. and Andrade, Á. G. Buffer transference strategy for power control in B5G-ultra-dense wireless cellular networks. Wirel. Netw. 28. 8. 3613-3620. 2022. https://doi.org/10.1007/s11276-022-03087-6

Pontencias en congresos:

- Anzaldo, A. and Andrade, Á. G. Interference-Aware Power Control for Spectrum Sharing Massive-IoT Communications. 14th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAml). 594. 2023. https://doi.org/10.1007/978-3-031-21333-5_46
- Anzaldo, A. and Andrade, Á. G. Deep Reinforcement Learning for Power control in Multi-tasks Wireless Cellular Networks. 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). 61-65. 2022. https://doi.org/10.1109/MeditCom55741.2022.9928617

- Anzaldo, A. and Andrade, Á. G. Training Effect on AI-based Resource Allocation in small-cell networks. 1-6. 2021. https://doi.org/10.1109/LATINCOM53176.2021.9647736

Otros:

- Coordinación de interferencia mediante control de potencia en redes móviles Ultra-Densas de nueva generación aplicando Aprendizaje por Refuerzo Profundo, trabajo presentado en el concurso institucional 3MT Tesis en tres minutos organizado por la Universidad Autónoma de Baja California.
- Asignación de Recursos Asistida por IA en redes Móviles de Nueva Generación, trabajo presentado en el 2do Seminario Virtual de Divulgación en Computación organizado por la Universidad Veracruzana. https://www.youtube.com/watch?v=9ly1hSNv318&ab_channel=C%C3%B3digoIA
- A Deep Reinforcement Learning based power coordination strategy for Ultra Dense Networks, trabajo presentado en IEEE ComSoc Industry – Student Panel organizado por IEEE Communications Society.

Referencias

[1] S. Chen, Y. Liang, S. Sun, S. Kang, W. Cheng and M. Peng. Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed. IEEE Wirel. Commun. 27. 2. 218-228. 2020.

[2] H. Viswanathan and P. E. Mogensen. Communications in the 6G Era. IEEE Access. 8. 57063-57074. 2020. https://doi.org/10.1109/ACCESS.2020.2981745

[3] Alsharif M.H., Kelechi A.H., Albreem, M.A., Chaudhry S.A., Zia, M.S. Kim, S. Sixth. Generation (6G) Wireless Networks: Vision, Research Activities, Challenges and Potential Solutions. Symmetry. 12. 4. 676. 2020. https://doi.org/10.3390/SYM12040676

[4] A. Yazar, S.Doğan Tusha, and H. Arslan. 6G vision: An ultra-flexible perspective. ITU Journal on Future and Evolving Technologies. 1. 1. 2020. https://doi.org/10.52953/IKVY9186

[5] M. Ding, D. Lopez-Perez, G. Mao, P. Wang and Z. Lin. Will the Area Spectral Efficiency Monotonically Grow as Small Cells Go Dense?. 2015 IEEE Global Communications Conference (GLOBECOM). 1-7. 2015. https://doi.org/10.1109/GLOCOM.2015.7416981

[6] L. Xu, Y. Mao, S. Leng, G. Qiao, Q. Zhao. Energy-efficient resource allocation strategy in ultra dense small-cell networks: A stackelberg game approach. 2017 IEEE International Conference on Communications (ICC). 2017. https://doi.org/10.1109/ICC.2017.7997289

[7] K. Shen and W. Yu. Fractional programming for communication systems—Part I: Power control and beamforming. IEEE Trans. Signal Process. 66. 10. 2616–2630. 2018. https://doi.org/10.1109/TSP.2018.2812733

[8] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. IEEE Trans. Signal Process. 59. 9. 4331–4340. 2011. https://doi.org/10.1109/TSP.2011.2147784

[9] Simon Atuah Asakipaam, Jerry John Kponyo, Kwame Oteng Gyasi. Resource Provisioning and Utilization in 5G Network Slicing: A Survey of Recent Advances, Challenges, and Open Issues. International Journal of Computer Networks and Applications. 10. 2. 201-216. 2023. https://doi.org/10.22247/ijcna/2023/220736

[10] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo and P. Soldati. Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges. IEEE Commun. Mag. 56. 9. 138-145. 2018. https://doi.org/10.1109/MCOM.2018.1701031

[11] Y. S. Nasir and D. Guo. Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks. IEEE J. Sel. Areas Commun. 37. 10. 2239-2250. 2019. https://doi.org/10.1109/JSAC.2019.2933973

[12] N. Naderializadeh, J. Sydir, M. Simsek and H. Nikopour. Resource Management in Wireless Networks via Multi-Agent Deep Reinforcement Learning. 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). 1-5. 2020. https://doi.org/10.1109/SPAWC48557.2020.9154250

[13] H. Li, T. Lv and X. Zhang. Deep Deterministic Policy Gradient Based Dynamic Power Control for Self-Powered Ultra-Dense Networks. 2018 IEEE Globecom Workshops (GC Wkshps). 1-6. 2018 https://doi.org/10.1109/GLOCOMW.2018.8644157

[14] Hui Ding, Feng Zhao, Jie Tian, Dongyang Li, Haixia Zhang. A deep reinforcement learning for user association and power control in heterogeneous networks. 102. 102069. 2020. https://doi.org/10.1016/J.ADHOC.2019.102069

[15] Xiao L., Zhang H., Xiao Y., Wan X., Liu S., Wang L. C., and Poor H. V. Reinforcement Learning-Based Downlink Interference Control for Ultra-Dense Small Cells. IEEE Trans. Wirel. 19. 1. 423-434. 2020. https://doi.org/10.1109/TWC.2019.2945951

[16] X. Liao, J. Shi, Z. Li, L. Zhang and B. Xia. A Model-Driven Deep Reinforcement Learning Heuristic Algorithm for Resource Allocation in Ultra-Dense Cellular Networks. IEEE Trans. Veh. 69. 1. 983-997. 2020. https://doi.org/10.1109/TVT.2019.2954538

[17] Liu X., Chen X., Chen Y., Li Z. Deep Learning Based Dynamic Uplink Power Control for NOMA Ultra-Dense Network System. Blockchain and Trustworthy Systems: First International Conference. 1156. 774-786. 2020. https://doi.org/10.1007/978-981-15-2777-7_64

[18] Q. Su, B. Li, C. Wang, C. Qin and W. Wang. A Power Allocation Scheme Based on Deep Reinforcement Learning in HetNets. 2020 International Conference on Computing, Networking and Communications (ICNC). 245-250. 2020. https://doi.org/10.1109/ICNC47757.2020.9049771

[19] S. Sritharan, H. Weligampola and H. Gacanin. A Study on Deep Learning for Latency Constraint Applications in Beyond 5G Wireless Systems. IEEE Access. 8. 218037-218061. 2020. https://doi.org/10.1109/ACCESS.2020.3040133

[20] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. https://doi.org/10.1038/NATURE14236

[21] Lin L-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine Learning. 8. 293–321. 1992. https://doi.org/10.1007/BF00992699

[22] Wang Kun, Mridul Aanjaneya, and Kostas Bekris. Sim2sim evaluation of a novel data-efficient differentiable physics engine for tensegrity robots. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2021. https://doi.org/10.1109/IROS51168.2021.9636783

[23] Dimitropoulos K., Hatzilygeroudis I., Chatzilygeroudis K. A Brief Survey of Sim2Real Methods for Robot Learning. Advances in Service and Industrial Robotics: RAAD. 133-140. 2022. https://doi.org/10.1007/978-3-031-04870-8_16

[24] Alam Saniul, Sadia Islam, Muhammad RA Khandaker, Risala T. Khan, Faisal Tariq, and Apriana Toding. Deep Q-Learning Based Resource Allocation in Interference Systems With Outage Constraint. 2022. https://doi.org/10.1109/VTCFALL.2019.8891446

[25] F. Meng, P. Chen and L. Wu. Power Allocation in Multi-User Cellular Networks with Deep Q Learning Approach. IEEE International Conference on Communications (ICC). 1-6. 2019. https://doi.org/10.1109/ICC.2019.8761431

[26] Xie Ronglei, Zhijun Meng, Lifeng Wang, Haochen Li, Kaipeng Wang, and Zhe Wu. Unmanned Aerial Vehicle Path Planning Algorithm Based on Deep Reinforcement Learning in Large-Scale and Dynamic Environments. IEEE Access. 9. 24884–24900. 2021. https://doi.org/10.1109/ACCESS.2021.3057485

[27] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. 2016. https://doi.org/10.48550/arXiv.1511.05952

[28] Wang Y, Zhang Z. Experience Selection in Multi-agent Deep Reinforcement Learning. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). 864–870. 2019. https://doi.org/10.1109/ICTAI.2019.00123

[29] Sharma A, Pal MK, Anand S, Kaul SK. Stratified Sampling Based Experience Replay for Efficient Camera Selection Decisions. 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM). 144–151. 2020. https://doi.org/10.1109/BIGMM50055.2020.00029

[30] Dao G, Lee M. Relevant Experiences in Replay Buffer. 2019 IEEE Symposium Series on Computational Intelligence (SSCI). 94–101. 2019. https://doi.org/10.1109/SSCI44817.2019.9002745

[31] de Bruin T, Kober J, Tuyls K, Babuška R. Improved deep reinforcement learning for robotics through distribution-based experience retention. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 3947–3952. 2016. https://doi.org/10.1109/IROS.2016.7759581

[32] B. Kitchenham. Procedures for performing systematic reviews. Keele. 33. 2004. 1–26. 2004.

[33] W. Yu, H. Xu, H. Zhang, D. Griffith, and N. Golmie. Ultra-dense networks: Survey of state of the art and future directions. 25th international conference on computer communication and networks (ICCCN). 1-10. 2016. https://doi.org/10.1109/ICCCN.2016.7568592

[34] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo. Machine learning paradigms for next-generation wireless networks. IEEE Wirel. Commun. 24. 2. 98–105. https://doi.org/10.1109/MWC.2016.1500356WC

[35] Y. Teng, M. Liu, F. R. Yu, V. C. Leung, M. Song, and Y. Zhang. Resource allocation for ultra-dense networks: A survey, some research issues and challenges. IEEE Commun. Surv. 21. 3. 2134–2168. 2018. https://doi.org/10.1109/COMST.2018.2867268

[36] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain. Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. IEEE Commun. Surv. 22. 2. 1251–1275. 2020. https://doi.org/10.1109/COMST.2020.2964534

[37] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao. Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surv. 21. 4. 3072–3108. 2019. https://doi.org/ 10.1109/COMST.2020.2964534

[38] N. C. Luong et al. Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys & Tutorials. 21. 4. 3133–3174. 2019. https://doi.org/10.1109/COMST.2019.2916583

[39] Q. V. Do and I. Koo. Actor-critic deep learning for efficient user association and bandwidth allocation in dense mobile networks with green base stations. Wireless Networks. 25. 5057–5068. 2019. https://doi.org/10.1007/s11276-019-02117-0

[40] L. Wang, C. Yang, X. Wang, J. Li, Y. Wang, and Y. Wang. Integrated resource scheduling for user experience enhancement: A heuristically accelerated drl. 11th International Conference on Wireless Communications and Signal Processing (WCSP). 1-6. 2019. https://doi.org/10.1109/WCSP.2019.8927970

[41] M. S. Hossain and G. Muhammad. A deep-tree-model-based radio resource distribution for 5G networks. IEEE Wireless Communications. 27. 1. 62–67. 2020. https://doi.org/10.1109/MWC.001.1900286

[42] H. Zhang et al. Distributed DNN Based User Association and Resource Optimization in mmWave Networks. IEEE Global Communications Conference (GLOBECOM). 1-5. 2019. https://doi.org/10.1109/GLOBECOM38437.2019.9014077

[43] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang. Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks. IEEE Transactions on Wireless Communications. 18. 11. 5141–5152. 2019. https://doi.org/10.1109/TWC.2019.2933417

[44] Z. Liu, X. Chen, Y. Chen, and Z. Li. Deep reinforcement learning based dynamic resource allocation in 5G ultra-dense networks. IEEE International Conference on Smart Internet of Things (SmartIoT). 168-174. 2019. https://doi.org/ 10.1109/SmartIoT.2019.00034

[45] Y. Zhou, Z. M. Fadlullah, B. Mao, and N. Kato. A deep-learning-based radio resource assignment technique for 5G ultra dense networks. IEEE Network. 32. 6. 28–34. 2018. https://doi.org/10.1109/MNET.2018.1800085

[46] A. Feki and V. Capdevielle. Autonomous resource allocation for dense lte networks: A multi armed bandit formulation. IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications. 66–70. 2011. https://doi.org/10.1109/PIMRC.2011.6140047

[47] J. Cao et al. Resource allocation for ultradense networks with machine-learning-based interference graph construction. IEEE Internet of Things Journal. 7. 3. 2137–2151. 2019. https://doi.org/10.1109/JIOT.2019.2959232

[48] M. Elsayed and M. Erol-Kantarci. Learning-based resource allocation for data-intensive and immersive tactile applications. IEEE 5G World Forum (5GWF). 278–283. 2018. https://doi.org/10.1109/5GWF.2018.8517001

[49] W. Lu, Q. Fan, Z. Li, and H. Lu. Power control based time-domain inter-cell interference coordination scheme in DSCNs. IEEE International Conference on Communications (ICC). 1–6. 2016. https://doi.org/10.1109/ICC.2016.7511467

[50] H. Zhang, M. Feng, K. Long, G. K. Karagiannidis, and A. Nallanathan. Artificial intelligence-based resource allocation in ultradense networks: Applying event-triggered Q-learning algorithms. IEEE Vehicular Technology Magazine. 14. 4. 56–63. 2019. https://doi.org/10.1109/MVT.2019.2938328

[51] W. AlSobhi and A. H. Aghvami. QoS-aware resource allocation of two-tier HetNet: A Q-learning approach. 26th International Conference on Telecommunications (ICT). 330–33. 2019. https://doi.org/10.1109/ICT.2019.8798829

[52] T. Jiang, Q. Zhao, D. Grace, A. G. Burr, and T. Clarke. Single-state Q-learning for self-organised radio resource management in dual-hop 5G high capacity density networks. Transactions on Emerging Telecommunications Technologies. 27. 12. 1628–1640. 2016. https://doi.org/10.1002/ETT.3019

[53] R. Amiri, M. A. Almasi, J. G. Andrews, and H. Mehrpouyan. Reinforcement learning for self organization and power control of two-tier heterogeneous networks. IEEE Transactions on Wireless Communications. 18. 8. 3933–3947. 2019. https://doi.org/10.1109/TWC.2019.2919611

[54] I. AlQerm and B. Shihada. Energy-efficient power allocation in multitier 5G networks using enhanced online learning. IEEE Transactions on Vehicular Technology. 66. 12. 11086–11097. 2017. https://doi.org/10.1109/TVT.2017.2731798

[55] H. Zhang, M. Min, L. Xiao, S. Liu, P. Cheng, and M. Peng. Reinforcement learning-based interference control for ultra-dense small cells. IEEE Global Communications Conference (GLOBECOM). 1–6. 2018. https://doi.org/10.1109/GLOCOM.2018.8648136

[56] D. Li, H. Zhang, K. Long, W. Huangfu, J. Dong, and A. Nallanathan. User association and power allocation based on Q-learning in ultra dense heterogeneous networks. IEEE Global Communications Conference (GLOBECOM). 1–5. 2019. https://doi.org/10.1109/GLOBECOM38437.2019.9013455

[57] Y. Li, Z. Gao, L. Huang, X. Du, and M. Guizani. Energy-aware interference management for ultra-dense multi-tier HetNets: Architecture and technologies Computer Communications. 127. 30–35. 2018. https://doi.org/10.1016/J.COMCOM.2018.05.012

[58] M. Chen, Y. Hua, X. Gu, S. Nie, and Z. Fan. A self-organizing resource allocation strategy based on Q-learning approach in ultra-dense networks. IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC). 155–160. 2016. https://doi.org/ https://doi.org/10.1109/ICNIDC.2016.7974555

[59] R. Amiri and H. Mehrpouyan. Self-organizing mm wave networks: A power allocation scheme based on machine learning. 11th Global symposium on millimeter waves (GSMM). 1–4. 2018. https://doi.org/10.1109/GSMM.2018.8439323

[60] S. Lin, J. Yu, W. Ni, and R. Liu. Radio resource management for ultra-dense smallcell networks: A hybrid spectrum reuse approach. IEEE 85th Vehicular Technology Conference (VTC Spring). 1–7. 2017. https://doi.org/10.1109/VTCSPRING.2017.8108229

[61] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak. A Machine Learning Approach for Power Allocation in HetNets Considering QoS. IEEE International Conference on Communications (ICC). 1–7. 2018. https://doi.org/ https://doi.org/10.1109/ICC.2018.8422864

[62] I. AlQerm and B. Shihada. A cooperative online learning scheme for resource allocation in 5G systems. IEEE International Conference on Communications (ICC). 1–7. 2016. https://doi.org/ https://doi.org/10.1109/ICC.2016.7511617

[63] M. Elsayed, M. Erol-Kantarci, B. Kantarci, L. Wu, and J. Li. Low-latency communications for community resilience microgrids: A reinforcement learning approach. IEEE Transactions on Smart Grid. 11. 2. 1091–1099. 2019. https://doi.org/10.1109/TSG.2019.2931753

[64] H. Khoshkbari, V. Pourahmadi, and H. Sheikhzadeh. Power allocation in cellular network without global csi: Bayesian reinforcement learning approach. 28th Iranian Conference on Electrical Engineering (ICEE). 1–6. 2020. https://doi.org/10.1109/ICEE50131.2020.9260583

[65] L. Li et al. Resource allocation for NOMA-MEC systems in ultra-dense networks: A learning aided mean-field game approach. IEEE Transactions on Wireless Communications. 20. 3. 1487–1500. 2020. https://doi.org/10.1109/ICCWORKSHOPS49005.2020.9145070

[66] Z. Li, X. Wen, Z. Lu, and W. Jing. A General DRL-based Optimization Framework of User Association and Power Control for HetNet. IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). 1141–1147. 2021. https://doi.org/10.1109/PIMRC50174.2021.9569426

[67] X. Huang, K. Zhang, F. Wu, and S. Leng. Collaborative machine learning for energy-efficient edge networks in 6G. IEEE Network. 35. 6. 12–19. 2021. https://doi.org/10.1109/MNET.100.2100313

[68] V. Vishnoi, P. K. Malik, I. Budhiraja, and A. Yadav. Deep Reinforcement Learning Based Throughput Maximization Scheme for D2D Users Underlaying NOMA-Enabled Cellular Network. Advanced Computing: 11th International Conference, IACC 2021, Msida, Malta, December 18–19. 318–331. 2021. https://doi.org/10.1007/978-3-030-95502-1_25

[69] Y. Li, X. Zhao, and H. Liang. Throughput maximization by deep reinforcement learning with energy cooperation for renewable ultradense IoT networks. IEEE Internet of Things Journal. 7. 9. 9091–9102. 2020. https://doi.org/10.1109/JIOT.2020.3002936

[70] Z. Cheng, M. LiWang, N. Chen, H. Lin, Z. Gao, and L. Huang. Learning-based joint user-AP association and resource allocation in ultra dense network. IEEE 91st Vehicular Technology Conference (VTC2020-Spring). 1–5. 2020. https://doi.org/10.1109/VTC2020-SPRING48590.2020.9128602

[71] Zhang H., Wang T., & Shen H. A Resource Allocation Algorithm for Ultra-Dense Networks Based on Deep Reinforcement Learning. International Journal of Computers Communications & Control. 16. 2. 2021. https://doi.org/ https://doi.org/10.15837/IJCCC.2021.2.4189

[72] M. M. Sande, M. C. Hlophe, and B. T. Maharaj. Access and radio resource management for IAB networks using deep reinforcement learning. IEEE Access. 9. 114218–114234. 2021. https://doi.org/10.1109/ACCESS.2021.3104322

[73] X. Chen, X. Liu, Y. Chen, L. Jiao, and G. Min. Deep Q-Network based resource allocation for UAV-assisted Ultra-Dense Networks. Computer Networks. 196. 108249. 2021. https://doi.org/10.1016/J.COMNET.2021.108249

[74] Y. Zhao, T. Peng, Y. Guo, and W. Wang. Energy-Efficient Uplink Power Allocation in Ultra-Dense Network Through Multi-agent Reinforcement Learning. IEEE 94th Vehicular Technology Conference (VTC2021-Fall). 1–7. 2021. https://doi.org/10.1109/VTC2021-FALL52928.2021.9625554

[75] Z. Ye and T. Ni. Intelligent Resource Allocation for Ultradense Networks Based on Improved Reinforcement Learning. Sci. Program. 2022. 2022. https://doi.org/10.1155/2022/9312847

[76] X. Zhang, Z. Zhang, and L. Yang. Learning-Based Resource Allocation in Heterogeneous Ultradense Network. IEEE Internet of Things Journal. 9. 20. 20229–20242. 2022. https://doi.org/10.1109/JIOT.2022.3173210

[77] Y. Li, Z. Tang, Z. Lin, Y. Gong, X. Du, and M. Guizani. Reinforcement Learning Power Control Algorithm Based on Graph Signal Processing for Ultra-Dense Mobile Networks. IEEE Transactions on Network Science and Engineering. 8. 3. 2694–2705. 2021. https://doi.org/10.1109/TNSE.2021.3051660

[78] M. U. Iqbal, E. A. Ansari, and S. Akhtar. Interference Mitigation in HetNets to Improve the QoS Using Q-Learning. IEEE Access. 9. 32405–32424. 2021. https://doi.org/10.1109/ACCESS.2021.3060480

[79] M. U. Iqbal, E. A. Ansari, S. Akhtar, and A. N. Khan. Improving the QoS in 5G HetNets Through Cooperative Q-Learning. IEEE Access. 10. 19654–19676. 2022. https://doi.org/10.1109/ACCESS.2022.3151090

[80] E. Kim, H.-H. Choi, H. Kim, J. Na, and H. Lee. Optimal Resource Allocation Considering Non-Uniform Spatial Traffic Distribution in Ultra-Dense Networks: A Multi-Agent Reinforcement Learning Approach. IEEE Access. 10. 20455–20464, 2022. https://doi.org/10.1109/ACCESS.2022.3152162

[81] J. Liu and H. Zhang. Power Allocation in Ultra-Dense Networks Through Deep Deterministic Policy Gradient. IEEE Wireless Communications Letters. 11. 12. 2502–2506. 2022. https://doi.org/10.1109/LWC.2022.3206096

[82] T. Chen et al. Efficient Uplink Transmission in Ultra-Dense LEO Satellite Networks With Multiband Antennas. IEEE Communications Letters. 26. 6. 1373–1377. 2022. https://doi.org/10.1109/LCOMM.2022.3160839

[83] H. Kim, J. So, and H. Kim. Carbon-Neutral Cellular Network Operation Based on Deep Reinforcement Learning. Energies. 15. 12. 4504. 2022. https://doi.org/10.3390/EN15124504

[84] K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju, and B. Shim. Deep Reinforcement Learning-Based Network Slicing for Beyond 5G. IEEE Access. 10. 7384–7395. 2022. https://doi.org/10.1109/ACCESS.2022.3141789

[85] N. Sharma and K. Kumar. Energy Efficient Clustering and Resource Allocation Strategy for Ultra-Dense Networks: A Machine Learning Framework. IEEE Transactions on Network and Service Management. 1–1. 2022. https://doi.org/10.1109/TNSM.2022.3218819

[86] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, and H. Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. IEEE Transactions on Neural Networks. 11. 3. 690–696. 2000. https://doi.org/10.1109/72.846740

[87] T. Hu and Y. Fei. QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks. IEEE Transactions on Mobile Computing. 9. 796–809. 2010. https://doi.org/10.1109/TMC.2010.28

[88] Y. Fu, S. Wang, C.-X. Wang, X. Hong, and S. McLaughlin, Artificial Intelligence to Manage Network Traffic of 5G Wireless Networks. IEEE Network. 32. 6. 58–64. 2018. https://doi.org/10.1109/MNET.2018.1800115

[89] C. Liu, J. Wang, X. Liu, and Y.-C. Liang. Deep CM-CNN for Spectrum Sensing in Cognitive Radio. IEEE Journal on Selected Areas in Communications. 37. 10. 2306–2321. 2019. https://doi.org/10.1109/JSAC.2019.2933892

[90] C. Zhang, P. Patras and H. Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys & Tutorials. 21. 3. 2224-2287. 2019. https://doi.org/10.1109/COMST.2019.2904897

[91] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah. Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial. IEEE Communications Surveys & Tutorials. 21. 4. 3039–3071. 2019. https://doi.org/10.1109/COMST.2019.2926625

[92] Hochreiter, Sepp, and Jürgen Schmidhuber. Long short-term memory. Neural computation. 9. 8. 1735-1780. 1997. https://doi.org/10.1162/neco.1997.9.8.1735
[93] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature. 521. 7553. 436-444. 2015. https://doi.org/10.1038/nature14539
[94] Kipf, Thomas N., and Max Welling. Semi-supervised classification with graph convolutional networks. 2016. https://doi.org/10.48550/arXiv.1609.02907
[95] Watkins, Christopher JCH, and Peter Dayan. Q-learning. Machine learning. 8. 279-292. 1992. https://doi.org/10.1007/BF00992698
[96] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. Dueling network architectures for deep reinforcement learning. International conference on machine learning. 1995–2003. 2016. https://doi.org/10.5555/3045390.3045601
[97] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. AAAI conference on artificial intelligence. 30. 1. 2016. https://doi.org/10.1609/AAAI.V30I1.10295
[98] T. Zhang and S. Mao. Energy-Efficient Power Control in Wireless Networks With Spatial Deep Neural Networks. IEEE Transactions on Cognitive Communications and Networking. 6. 1. 111–124. 2020. https://doi.org/10.1109/TCCN.2019.2945774
[99] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
[100] A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. Machine learning. 13. 103–130. 1993. https://doi.org/10.1007/BF00993104
[101] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M.G. Bellemare, A.Graves, M.R iedmiIler, A. K.Fi djeland, and G. Ostrovski. Human-l evel controlth rough deep reinforcementl earning. Nature. 518. 7540. 529-533. 2015. https://doi.org/10.1038/NATURE14236
[102] Sutton, R. S., & Barto, A. G. Reinforcement learning: An introduction. MIT press. 2018. https://doi.org/10.1016/S0893-6080(99)00098-2
[103] Hasselt, H. Double Q-learning. Advances in neural information processing systems. 23. 2010. https://doi.org/10.7551/MITPRESS/1120.003.0006
[104] Pan, S. J., & Yang, Q. A survey on transfer learning. IEEE Transactions on knowledge and data engineering. 22. 10. 1345-1359. 2010. https://doi.org/10.1109/TKDE.2009.191
[105] Z. Yang and K. L. Yeung. SDN candidate selection in hybrid IP/SDN networks for single link failure protection. IEEE/ACM Trans. Netw. 28. 1. 312–321. 2020. https://doi.org/10.1109/TNET.2019.2959588
[106] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. IEEE Trans. Signal Process. 66. 20. 5438–5453. 2018. https://doi.org/10.1109/TSP.2018.2866382
[107] F. Meng, P. Chen, L. Wu, and J. Cheng. Power allocation in multiuser cellular networks: Deep reinforcement learning approaches. IEEE Trans. Wireless Commun. 19. 10. 6255–6267. 2020. https://doi.org/10.1109/TWC.2020.3001736
[108] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li. Spectrum and power allocation for vehicular communications with delayed CSI feedback. IEEE Wireless Commun. Lett. 6. 4. 458–461. 2017. https://doi.org/10.1109/LWC.2017.2702747
[109] Wang, M., Gao, H., & Lv, T. Energy-efficient user association and power control in the heterogeneous network. IEEE Access. 5. 5059–5068. https://doi.org/10.1109/ACCESS.2017.2690305
[110] Zhang, L., & Liang, Y. C. Deep reinforcement learning for multi-agent power control in heterogeneous networks. IEEE Transactions on Wireless Communications. 20. 4. 2551-2564. 2020. https://doi.org/10.1109/GLOBECOM42002.2020.9322443
[111] Zhao, Y., Niemegeers, I. G., & De Groot, S. M. H. Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods. IEEE Access. 9. 102953-102965. 2021. https://doi.org/10.1109/ACCESS.2021.3097243
[112] Naparstek, O., & Cohen, K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. IEEE Trans. Wirel. 18. 1. 310-323. 2018. https://doi.org/10.1109/TWC.2018.2879433
[113] Huang, J., Berry, R. A., & Honig, M. L. Distributed interference compensation for wireless networks. IEEE Journal on Selected Areas in Communications. 24. 5. 1074-1084. 2016. https://doi.org/10.1109/JSAC.2006.872889
[114] S. Zhang and R. S. Sutton. A deeper look at experience replay. 2017. https://doi.org/10.24963/IJCAI.2018/666

[115] Nicholaus, I. T., & Kang, D. K. Robust experience replay sampling for multi-agent reinforcement learning. Pattern Recognition Letters. 155. 135-142. 2022. https://doi.org/10.1016/J.PATREC.2021.11.006

[116] Da Silva, F. L., & Costa, A. H. R. A survey on transfer learning for multiagent reinforcement learning systems. Journal of Artificial Intelligence Research. 64. 645-703. 2019. https://doi.org/10.1613/JAIR.1.11396

[117] Dong, T., Qi, Q., Wang, J., Liu, A. X., Sun, H., Zhuang, Z., & Liao, J. Generative adversarial network-based transfer reinforcement learning for routing with prior knowledge. IEEE Transactions on Network and Service Management. 18. 2. 1673-1689. 2021. https://doi.org/10.1109/TNSM.2021.3077249

[118] Chan, S. C., Fishman, S., Canny, J., Korattikara, A., & Guadarrama, S. Measuring the reliability of reinforcement learning algorithms. 2019. https://doi.org/10.22541/AU.149693987.70506124

APÉNDICE A: PRUEBAS DE HIPÓTESIS

En este trabajo, se utilizó la prueba t de Student con un nivel de significancia de 0.05 para evaluar si existían diferencias significativas en los tiempos transitorios y la variabilidad de la capacidad de la red durante el entrenamiento de ajuste, al considerar los mecanismos de gestión propuestos. Las variantes de los mecanismos propuestos, incluyendo la reutilización de experiencias adquiridas bajo diversas condiciones (EIT) y el mecanismo de gestión de experiencias que retiene las experiencias en dos buffers independientes (DER o FDER), fueron comparados con las configuraciones sin los mecanismos propuestos (NIT y UER).

A continuación, se presentan las hipótesis planteadas junto con sus respectivas tablas de resultados.

- Hipótesis nula 1 ($H1_0$): La combinación de un mecanismo de gestión de experiencias y la reutilización de experiencias adquiridas bajo diversas condiciones (ubicación de UE y CSI) no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia en redes móviles B5G.
- Hipótesis alternativa 1 ($H1_1$): La combinación de un mecanismo de gestión de experiencias y la reutilización de experiencias adquiridas bajo diversas condiciones (ubicación de UE y CSI) reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia en redes móviles B5G.

Experimento 1 (TI:10)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
UER-1UE	1650	2525	10	-0.7504	0.2372	No Rechazar
PER-1UE	1800	2575	10	-6.037	0.0002	Rechazar
CER-1UE	1650	1375	10	0.4146	0.6553	No Rechazar
DER-1UE	1575	1050	10	0.4846	0.6795	No Rechazar
UER-4UE	1400	5000	10	-6.6712	7.87E-05	Rechazar
PER-4UE	375	5000	10	-5.0097	0.0005	Rechazar
CER-4UE	425	2475	10	-2.1222	0.0333	Rechazar
DER-4UE	1350	1375	10	-0.0274	0.4894	No Rechazar
Experimento 1 (TI:1)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
UER-1UE	750	550	10	0.2413	0.5923	No Rechazar

PER-1UE	1150	600	10	0.8287	0.7844	No Rechazar
CER-1UE	650	300	10	1.0853	0.8453	No Rechazar
DER-1UE	800	275	10	0.6279	0.7262	No Rechazar
UER-4UE	1150	4100	10	-4.0294	0.0018	Rechazar
PER-4UE	1200	5000	10	-4.2991	0.0013	Rechazar
CER-4UE	350	5000	10	-12.2675	9.06E-07	Rechazar
DER-4UE	1200	1625	10	-0.3821	0.3562	No Rechazar
Experimento 2	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
FUER	2300	2000	10	0.597614	0.7166	No Rechazar
FDER	1600	1300	10	1.06066	0.8400	No Rechazar
Experimento 4 (25 BS)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
EIT1	36840	32920	50	2.448012	0.9919	No Rechazar
EIT2	31280	32920	50	-0.86427	0.1947	No Rechazar
EIT3	24540	32920	50	-3.76697	0.0001	Rechazar
Experimento 4 (36 BS)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
EIT1	35340	28860	50	3.928141	0.9999	No Rechazar
EIT2	29160	28860	50	0.182917	0.5723	No Rechazar
EIT3	15920	28860	50	-6.52382	1.51E-09	Rechazar
EIT4	8040	28860	50	-9.78773	1.75E-16	Rechazar

- Hipótesis nula 2 ($H2_0$): La implementación de un mecanismo de gestión de experiencias que retenga las experiencias en dos buffers independientes (de exploración y de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia.
- Hipótesis alternativa 2 ($H2_1$): La implementación de un mecanismo de gestión de experiencias que retenga las experiencias en dos buffers independientes (de exploración y de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G reducirá el tiempo transitorio de los modelos Deep Q-Network durante la asignación de potencia.

Experimento 1 (TI:10)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER-EIT-1UE	1575	1650	10	-4.1926	0.0015	Rechazar
DER-DIT-1UE	1525	1575	10	-0.5176	0.3094	No Rechazar
DER-NIT-1UE	1050	2525	10	-1.6653	0.0672	No Rechazar
DER-EIT-4UE	1350	1400	10	-0.0669	0.4741	No Rechazar
DER-DIT-4UE	2225	5000	10	-3.4283	0.0044	Rechazar
DER-NIT-4UE	1375	5000	10	-4.8119	0.0006	Rechazar

Experimento 1 (TI:1)	Media propuesta	Media control	Número de muestras	valor t	valor p	Resultado
DER-EIT-1UE	800	750	10	0.2348	0.5899	No Rechazar
DER-DIT-1UE	675	1150	10	-2.3691	0.0227	Rechazar
DER-NIT-1UE	275	550	10	-2.4448	0.0201	Rechazar
DER-EIT-4UE	1200	1150	10	0.0476	0.51839	No Rechazar
DER-DIT-4UE	3150	4650	10	-2.0263	0.03865	Rechazar
DER-NIT-4UE	1650	4100	10	-2.994	0.00861	Rechazar
Experimento 2	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER	1300	2000	10	-2.09165	0.03491	Rechazar
FDER	1600	2300	10	-1.49241	0.0869	No Rechazar
Experimento 3	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER- 1UE-10K	1000	13500	10	-4.81899	6.89E-05	Rechazar
DER-1UE-50K	1500	2000	10	-3.93279	0.0004	Rechazar
DER-4UE-10K	10500	30000	10	-10.355	2.60E-09	Rechazar
DER-4UE-50K	13500	30000	10	-8.91924	2.52E-08	Rechazar

- Hipótesis nula 3 ($H3_0$): La implementación de un mecanismo de gestión de experiencias que retenga experiencias en dos buffers independientes (i.e., buffer de exploración y buffer de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G no reducirá la variación de la capacidad de la red durante la asignación de potencia en los modelos Deep Q-Network.
- Hipótesis alternativa 3 ($H3_1$): La implementación de un mecanismo de gestión de experiencias que retenga experiencias en dos buffers independientes (i.e., buffer de exploración y buffer de explotación) durante el aprendizaje de nuevos entornos de redes móviles B5G reducirá la variación de la capacidad de la red durante la asignación de potencia en los modelos Deep Q-Network.

Experimento 1 (TI:10)	Media propuesta	Media control	Número de muestras	valor t	valor p	Resultado
DER-EIT-1UE	0.1085	0.1134	1600	-1.5607	0.05935	No Rechazar
DER-DIT-1UE	0.1181	0.1316	1600	-4.1394	1.78E-05	Rechazar
DER-NIT-1UE	0.1028	0.1067	1600	-1.5048	0.0662	No Rechazar
DER-EIT-4UE	0.0778	0.0876	1600	-4.7019	1.34E-06	Rechazar
DER-DIT-4UE	0.0813	0.1084	1600	-10.7427	9.02E-27	Rechazar
DER-NIT-4UE	0.0606	0.1081	1600	-25.1986	2.9E-128	Rechazar
Experimento 1 (TI:1)	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER-EIT-1UE	0.0874	0.1143	1600	-8.2989	7.11E-08	Rechazar

DER-DIT-1UE	0.1026	0.1030	1600	-0.1186	0.45281	No Rechazar
DER-NIT-1UE	0.1038	0.0999	1600	1.4724	0.9294	No Rechazar
DER-EIT-4UE	0.0858	0.1017	1600	-8.2688	9.81E-17	Rechazar
DER-DIT-4UE	0.0812	0.0721	1600	5.4885	1	No Rechazar
DER-NIT-4UE	0.0774	0.0716	1600	3.5538	0.9998	No Rechazar
Experimento 2	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER	0.8601	0.8303	30	0.3905	0.6512	No Rechazar
FDER	0.6900	0.6716	30	0.3072	0.6201	No Rechazar
Experimento 3	**Media propuesta**	**Media control**	**Número de muestras**	**valor t**	**valor p**	**Resultado**
DER-1UE-10K	0.0538	0.1599	61	-6.6845	3.86E-10	Rechazar
DER-1UE-50K	0.0547	0.0742	61	-2.2006	0.01483	Rechazar
DER-4UE-10K	0.1291	0.2301	61	-3.7966	0.0001	Rechazar
DER-4UE-50K	0.0880	0.2344	61	-5.1444	5.3E-07	Rechazar