DESARROLLO DE UN ALGORITMO DE APRENDIZAJE POR REFUERZO PROFUNDO PARA RESOLVER EL DESPACHO HIDROTÉRMICO COLOMBIANO CONSIDERANDO ESCENARIOS HIDROLÓGICOS Y DE DEMANDA BAJO INCERTIDUMBRE.

ALEJANDRO RAMÍREZ ARANGO

Proyecto de grado para optar al título de Magister en Ciencia de los Datos y Analítica

Director
Ph.D JOSE LISANDRO AGUILAR CASTRO

UNIVERSIDAD EAFIT
MAESTRIA EN CIENCIA DE LOS DATOS Y ANALÍTICA
MEDELLIN
2022

Contenido

El despacho económico es un problema de optimización ampliamente analizado en el sector eléctrico, que busca hacer el mejor uso de los recursos disponibles para satisfacer la demanda a mínimo costo. Este problema presenta un gran reto en su solución debido a la incertidumbre de múltiples parámetros, como la demanda de energía eléctrica, y para el caso colombiano es de especial interés la incertidumbre hidrológica por su alta dependencia en centrales hidroeléctricas. Dado que el despacho económico se asemeja a un problema de decisiones secuenciales, es posible modelar el problema como un proceso de decisión de Markov, lo que permite incorporar en la modelación la incertidumbre de los parámetros de interés. El presente proyecto propone una modelación del modelo de despacho económico colombiano como un proceso de decisión de Markov, considerando la incertidumbre en la demanda y la hidrología. Luego, a través de algoritmos de aprendizaje reforzado profundo se determina una política óptima y robusta para dar un mejor manejo a los recursos disponibles frente al manejo de la demanda energética.

1.1. PLANTEAMIENTO DEL PROBLEMA.

El despacho económico es un problema de optimización ampliamente utilizado en el sector eléctrico enfocado en la toma de decisiones para hacer el mejor uso de los recursos disponibles y satisfacer las necesidades de demanda a mínimo costo [1]. Sin embargo, para asegurar el mejor uso de los recursos disponibles se requiere de algo más que minimizar el costo de producción de generación de las centrales eléctricas, dado que se deben tener en cuenta diversos factores, tales como: variación de la demanda, fallas en la red, fallas en las centrales, restricciones ambientales, condiciones hidrológicas, radiación solar, velocidad del viento, limitaciones de combustible, reservas de seguridad, entre otras.

La complejidad del problema aumenta cuando es necesario realizar proyecciones porque, entre otros factores, para las centrales hidroeléctricas se introduce una dependencia temporal entre la decisión operativa de hoy y los costos operativos en el futuro [2], debido a la posibilidad de almacenar energía mediante embalses y atender requerimientos de demanda futuros donde la operación con otras plantas no es tan económica, lo que introduce el concepto de costo de oportunidad del agua.

Una proyección del despacho puede ser resuelto de manera determinística [3] utilizando optimizadores comerciales. Sin embargo, es claro que este enfoque no es óptimo dado que supone información perfecta sobre el futuro, por ejemplo: la hidrología en los meses siguientes. Este supuesto se vuelve más crítico para el caso colombiano donde las plantas hidroeléctricas tienen altos niveles de participación en la generación de energía, siendo en mayo de 2021 del 85.93% [4]. Sin embargo, considerar diferentes escenarios de hidrología hace que el modelo sufra la conocida maldición de dimensionalidad [5]. Aunque existe una

amplia diversidad de metodologías en la literatura para abordar una versión estocástica del despacho [3, 5, 6, 7], sigue siendo un tema en investigación por su alta complejidad.

Por otro lado, recientemente se han propuesto algoritmos de aprendizaje por refuerzo (Reinforcement Learning - RL) a un amplio campo de aplicaciones, por ejemplo: para el control del péndulo invertido, para juegos de computadora (eg. Tick and Toe, Backgammon, breakout, GO y space invaders) [8, 9, 10], para el control óptimo de sistemas [11], en la modelación de mercados de energía mayorista [12, 13] por mencionar algunos, donde se ha evidenciado tener buenos resultados para el manejo de variables aleatorias y decisiones secuenciales.

Considerando la amplia gama de aplicaciones de RL, surge la inquietud en verificar la aplicabilidad de estos algoritmos en el contexto del mercado de energía mayorista colombiano, incorporando en la modelación la incertidumbre en la demanda y la hidrología. Por lo anterior, surge la siguiente pregunta de investigación: ¿cuáles son los factores que se deben considerar para desarrollar un algoritmo de RL que permita resolver el problema de despacho económico para un horizonte de tiempo dado para el caso colombiano?

1.2. JUSTIFICACIÓN.

Contar con un modelo que permita realizar proyecciones del sistema de generación de energía eléctrica es importante, porque permite mejorar la toma de decisiones sobre precios de bolsa, planeación del sistema eléctrico, análisis de riesgo de desabastecimiento por efecto climático, proyecciones financieras, estimación de precios de contratación de energía de largo plazo, entre otros [1, 14, 15]. En este sentido, el modelo es una herramienta importante para los actores y reguladores del sector eléctrico.

El modelo de despacho hidrotérmico es generalmente la base para todos los análisis descritos, y aunque existen diversos métodos para resolver este tipo de problemas, la complejidad al considerar la incertidumbre en variables o el impacto en las decisiones operativas de cuánto generar, hace que generalmente se analice cada punto por separado para simplificar el problema. Dado que el RL es un marco general para resolver problemas que implican la toma de decisiones secuenciales, y que permite el manejo de la incertidumbre en el entorno [16], se observa un potencial interesante en su aplicación en la solución del despacho hidrotérmico de manera general, debido a que se puede enmarcar como un problema que implica la toma de decisiones siguiendo un modelo markoviano.

Así, con la ejecución de este proyecto, se busca contribuir en el campo de aprendizaje por refuerzo profundo mediante su aplicación particular al problema del despacho energético, estableciendo los factores generales que se deben considerar para incluir en la formulación la incertidumbre hidrológica, en conjunto con las decisiones operativas de generación, que es un punto que en la literatura se ha tratado poco.

1.3. OBJETIVOS.

1.3.1. Objetivo General.

- Desarrollar un algoritmo de aprendizaje profundo para el problema de despacho de energía colombiano hidro-térmica considerando escenarios hidrológicos y de demanda bajo incertidumbre.

1.3.2. Específicos.

- Caracterizar y modelar el problema de despacho económico a mínimo costo para el caso colombiano considerando diferentes escenarios de hidrología y de demanda.

- Definir el problema de despacho económico hidro-térmico como un proceso de decisión de Markov (MDP) bajo incertidumbre.

- Establecer varias alternativas de resolución del problema usando diversas estrategias de aprendizaje por reforzamiento profundo.

CAPÍTULO 2: MARCO CONCEPTUAL Y ESTADO DEL ARTE.

2.1. MARCO CONCEPTUAL.

2.1.1. Fundamentos de Optimización:

La optimización consiste en un proceso de búsqueda de la mejor solución, entre todas las posibles, para un problema dado [17]. La optimización cuenta con un amplio campo de aplicación en diversos sectores, tales como: finanzas, producción, distribución, localización, inventarios y asignación, por mencionar algunos. Los elementos que componen un modelo de optimización son: variables de decisión, función objetivo y restricciones. De acuerdo con el dominio de las variables y la linealidad de las funciones, los modelos de optimización se clasifican en los siguientes tipos [17, 25]:

- **Programación Lineal:** Se caracteriza porque el dominio de las variables de decisión son números reales positivos, y la linealidad de la función objetivo y las restricciones. Algunos algoritmos utilizados para solucionar estos problemas son: simplex, simplex dual, punto interior y descomposición de Danzing-Wolfe.

- **Programación No Lineal:** Se caracteriza porque la función objetivo y las restricciones son no lineales. Entre los algoritmos para solucionar estos problemas, según la complejidad, se tiene: Programación no restringida, programación cuadrática, relajación lagrangiana, condiciones de Karush-Kuhn-Tucker y programación separable.

- **Programación Entera:** Se diferencia de la programación lineal en que la variable de decisión toma únicamente valores enteros. Entre los algoritmos para solucionar estos problemas se tiene: branch and bound, enumeración explícita, planos cortantes y método húngaro.

- **Programación Estocástica:** Este tipo de problemas considera que algunos de los parámetros del modelo matemático son variables aleatorias, con o sin distribución de probabilidad conocida. Entre los algoritmos para solucionar estos problemas se tiene: descomposición de Benders, wait and see, here and now y simulación.

- **Metaheurísticos:** Son una clase de métodos aproximados diseñados para resolver problemas difíciles en los que los algoritmos antes señalados no son efectivos y eficientes. Entre los algoritmos para solucionar estos problemas se tienen: búsqueda tabú, recocido simulado, algoritmos genéticos, búsqueda dispersa y colonia artificial de hormigas.

2.1.2. Aprendizaje Automático:

El aprendizaje automático (Machine Learning - ML) es un área de la inteligencia artificial (Artificial Intelligence - AI) que permite a un sistema aprender de los datos o de la experiencia, entre otras fuentes [46]. Un modelo de conocimiento (por ejemplo, de predicción, de descripción, de optimización, etc.) es la salida que se genera cuando se entrena un algoritmo de aprendizaje automático. Entre los paradigmas de aprendizaje que existen en el aprendizaje automático se tienen [46]:

Aprendizaje supervisado: Parte de un conjunto conocido de datos y una cierta comprensión de cómo se clasifican estos datos. Estos datos tienen características etiquetadas que definen el significado de los datos. Su aplicación se centra en la regresión y clasificación.

Aprendizaje no supervisado: A diferencia del aprendizaje supervisado, el aprendizaje no supervisado parte de una cantidad masiva de datos que no tienen características etiquetadas. Este aprendizaje lleva a cabo un proceso iterativo buscando similaridades en los datos, sin intervención humana. Su aplicación principal es el agrupamiento.

2.1.3. Aprendizaje por Refuerzo:

El aprendizaje por refuerzo (Reinforcement Learning - RL) es otro paradigma de ML, y consiste en determinar cómo un agente interactúa con el entorno a través de acciones para obtener la máxima recompensa total acumulada (retorno) [18]. Los principales elementos de un sistema de RL son: política, señal de recompensa, función de valor, y de manera opcional un modelo del entorno. La *política* determina el comportamiento del agente en un determinado estado. La *señal de recompensa* define la meta que se busca lograr con el problema de RL. Mientras que la señal de recompensa indica lo que es bueno a corto plazo, la *función de valor* indica lo que es bueno a largo plazo. Finalmente, el *modelo del entorno* permite realizar inferencias sobre el posible comportamiento del entorno usando el modelo. Los modelos del entorno se utilizan para hacer planeación de las acciones, y son llamados métodos basados en modelo (model-based), mientras que los que están basados en ensayo y error son llamados métodos sin modelo (model-free). En general, un algoritmo de RL busca maximizar la recompensa total que se recibe [8].

Los algoritmos de RL están fundamentados en los procesos de decisión de Markov (*Markov Decision Procces - MDP*), definidos por la tupla $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma)$ donde $\mathcal{S}$ es un conjunto finito de estados $S_t \in \mathcal{S}$, $\mathcal{A}$ es un conjunto finito de acciones $A_t \in \mathcal{A}(s)$, P es la función de probabilidad de transición, tal que $P: \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$, donde $\Delta(\mathcal{S})$, es la distribución sobre los estados, $\mathcal{R}$ es la función de recompensa $R_t \in \mathcal{R} \subset \mathbb{R}$, y γ es un factor de descuento $\gamma \in (0,1)$ [8]. La interacción entre un agente y un entorno presenta la siguiente trayectoria [19]:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3 \ldots$$

Esta interacción puede resumirse en la Figura 1.

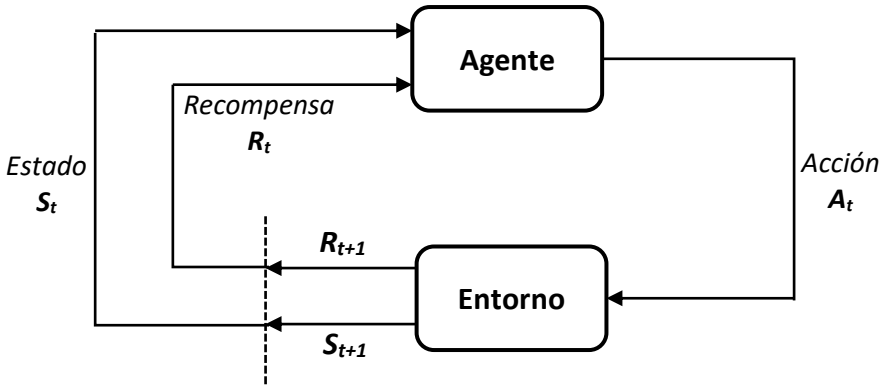


Figura 1. Representación básica de los elementos que involucran un modelo de RL. Fuente: [8]

Un supuesto importante en un MDP es que la probabilidad de cada posible valor de S_t y R_t depende solo de la acción y el estado inmediatamente anterior, S_{t-1} y A_{t-1}. El estado debe incluir información acerca de las interacciones pasadas entre el agente y el entorno. Cuando lo anterior se cumple, se dice que el estado tiene la propiedad de Markov [8].

En general, los métodos de RL buscan estimar la función de valor para estimar que tan bueno es estar en un estado dado. La notación de bueno está definida en función de las recompensas futuras esperadas (retorno esperado), que dependen de las acciones tomadas. Como se dijo antes, la manera o estrategia de elegir las acciones es conocido como política π, que formalmente es un mapeo de estados a probabilidades de seleccionar cada acción posible. Si un agente sigue la política π en el tiempo t, entonces $\pi(a|s)$ es la probabilidad de que $A_t = a$ si $S_t = s$ [8].

Una política π se define como la función estado-valor (*state-value function*) $v_\pi(s)$, como la recompensa esperada acumulada con descuento o retorno esperado [16]:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} |S_t = s, A_t = a\right] \quad (1)$$

$$\mathrm{v}_\pi(\mathrm{s}) \;=\; \mathbb{E}_s^\pi\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t)\right] \quad (2)$$

Resolver un problema de RL consiste en encontrar una política que logre la mayor recompensa a largo plazo [16], o, en otros términos, que satisfaga la función de estado-valor óptima $v_*(s) = \max_\pi v_\pi(s)$ definida como:

$$v_*(s) = \max_{\mathrm{a}}\left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v_*(s')\right] \quad (3)$$

De manera similar se define la función acción-valor $q_\pi(s,a)$, que describe el retorno esperado de tomar la acción a en el estado *s* siguiendo la política π:

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v_\pi(s') \tag{4}$$

Finalmente, la función óptima de acción-valor $q_*(s,a)$ es el valor óptimo que se alcanza al tomar la acción a:

$$q_*(s,a) = r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v_*(s') \tag{5}$$

En RL, cuando no se tiene conocimiento completo del entorno, es necesario estimar la función de valor para descubrir la política óptima. Entre los métodos para lograr lo anterior se encuentran [8]:

- **Monte Carlo:** Es un método para resolver problemas de RL basado en promediar los retornos de la muestra. Se asume que la experiencia está divida en episodios, y que eventualmente termina sin importa las acciones que se seleccionen. Solo al final de un episodio se estiman valores de la función de valor y cambia la política.

- **Temporal-Difference Learning (TD):** Es una combinación entre las ideas de Monte Carlo y Programación Dinámica. El aprendizaje de este método es a través de la experiencia, sin necesidad de tener conocimiento completo de la dinámica del entorno, por lo cual estima según lo aprendido en otras estimaciones. De acuerdo con el problema a resolver y el enfoque de exploración y explotación, los métodos más importantes son: Sarsa, Q-learning, Expected Sarsa, entre otros.

2.1.4. Aprendizaje por Refuerzo Profundo:

Cuando la función acción-valor se construye para un espacio discreto de baja dimensión, es posible almacenar el valor q para cada par estado-acción. Sin embargo, esta situación es poco usual en problemas prácticos donde generalmente el espacio de exploración es de alta dimensionalidad, o de estados continuos. Para manejar problemas de alta dimensión, se introduce la función de aproximación $\hat{q}_\theta = \hat{q}_\theta(s,a,\theta)$, donde $\theta \in \mathbb{R}^M$ es un vector que parametriza la función de aproximación, tal que se busca obtener acciones similares para estados similares. Debido a la capacidad de las redes neuronales para extraer características complejas, es una herramienta útil para calcular el parámetro θ, y, por tanto, abordar problemas con estados continuos o de altas dimensiones. El aprendizaje por refuerzo profundo es un método específico del RL que consiste en combinar las redes neuronales con algoritmos de RL para aprender directamente de datos de altas dimensiones [44]. Los esquemas clásicos de RL por refuerzo son para aprender la función de valor o la política. En el caso de la primera, se deriva una política a través de acciones ambiciosas que maximizan

la función valor con el mayor retorno esperado [18]. En el caso de la segunda, consiste en optimizar directamente la función valor para una política dada que maximice el retorno esperado [18]. Esquemas más avanzados combinan los métodos basados en valor y política. El presente proyecto aborda dos métodos de aprendizaje por refuerzo profundo: *Deep Q Network* y *Advance Actor Critic,* los cuáles se describen a continuación.

2.1.5. Aleatorio (Random):

El método aleatorio es una estrategia utilizada frecuentemente como caso de control en RL [10], donde se simula el caso cuando no hay aprendizaje por parte del agente [20]. Lo anterior sirve para establecer un equivalente a cero, y tener una base de comparación del nivel de aprendizaje o desempeño por parte de los algoritmos de RL que se están analizando.

El algoritmo aleatorio se detalla en el Anexo II (ver tabla 5), donde básicamente la estrategia del agente consiste en seleccionar acciones aleatorias dentro del dominio del entorno. Esta acción se aplica al entorno, el cual retorna un nuevo estado y una recompensa, que el agente en este caso no aprovecha para el aprendizaje. Sin embargo, según el problema, es posible plantear una variante a este algoritmo, donde el agente memoriza la acción que mayor retorno acumulado genera (ver paso 12 del algoritmo de la tabla 5), con el objetivo de validar si el entorno es lo suficientemente complejo para no ser resuelto bajo esta simple estrategia.

2.1.6. Deep Q Network - DQN:

Q-learning es un método basado en valor que estima el retorno esperado de la función $Q(S_t, a_t)$ a través de un retorno estimado $R_{t+1} + \gamma \cdot argmaxQ(S_{t+1}, a)$ conocido como "***TD-target***" [16]. El objetivo de Q-learning es estimar los valores de una política óptima seleccionando la acción que retorne el mayor valor esperado de la función Q-value para cada estado visitado [16]. Para un paso de Q-learning, la regla de actualización es la siguiente:

$$Q^{n+1}(s_t, a_t) = Q^n(\mathrm{s}_t, a_t) + \alpha \cdot \left[R(\mathrm{s}_t, a_t, s_{t+1}) + \gamma \cdot \max_{a \in A_{t+1}} Q^n(\mathrm{s}_{t+1}, a) - Q^n(\mathrm{s}_t, a_t)\right] \quad (6)$$

Donde A es el conjunto de acciones, $R(s_t, a_t, s_{t+1})$ es la función de recompensa, α es el parámetro de aprendizaje, γ es el factor de descuento, s_t representa el estado actual y s_{t+1} el estado siguiente al estado s_t al ejecutar la acción a_t, y $Q^n(\mathrm{s}_{t+1}, a)$ es la mejor estimación de *Q-value* para una acción a tomada en el estado s_{t+1}.

En un problema de altas dimensiones donde no es posible mapear cada par estado-acción de la función Q, se utilizan funciones de aproximación. Las redes neuronales son un tipo de

función de aproximación que aplicado en *Q-learning* conforman lo que se conoce como *Deep-Q-Network* (DQN).

El algoritmo DQN consiste en utilizar dos redes neuronales para aproximar la función $Q(s_t, a_t)$. Estas redes neuronales conocidas como *Target Network* y *Q-network* buscan estimar la recompensa futura y el valor de la función Q, respectivamente. Se denota como *Q-network* y *Target-network* a θ^Q y $\theta^{Q'}$, respectivamente. La Figura 2 ilustra el rol que cada red neuronal cumple en la regla de actualización de DQN:

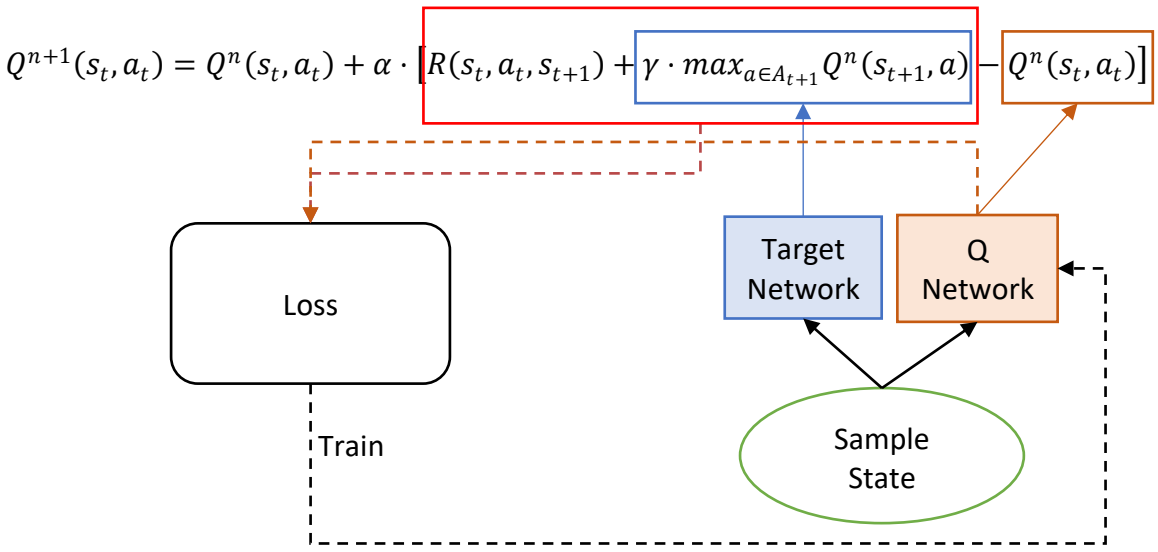


Figura 2. Esquema general que muestra como las redes neuronales en DQN se utilizan para aproximar la regla de actualización. Fuente: Elaboración propia

De acuerdo con la Figura 2, $\theta^{Q'}$ estima la recompensa futura por tomar una acción a. Al sumar esta estimación por la recompensa se obtiene como resultado una aproximación de la recompensa en t. Reescribiendo $R(s_t, a_t, s_{t+1}) + \gamma \cdot max_{a \in A_{t+1}} Q^n(s_{t+1}, a)$, considerando la red neuronal, se tiene la siguiente expresión equivalente:

$$R(s_t, a_t, s_{t+1}) + \gamma \cdot max_{a \in A_{t+1}} \theta^{Q'} \quad (7)$$

La predicción de θ^Q y la suma entre la recompensa y la predicción de $\theta^{Q'}$ se utilizan como argumentos para calcular el error a través de una función de pérdida (*Loss*). Suponiendo que la función de pérdida es la ecuación de mínimos cuadrados (MSE), el cálculo del error tendría la siguiente forma:

$$\mathcal{L}(\theta) = \left[\left(R(s_t, a_t, s_{t+1}) + \gamma \cdot argmaxQ(s', a'; \theta') - Q(s, a; \theta)\right)^2\right] \quad (8)$$

O de otra forma:

$$\mathcal{L}(\theta) = \left[\left(R(s_t, a_t, s_{t+1}) + \gamma \cdot \theta^{Q'} - \theta^{Q}\right)^2\right] \quad (9)$$

$$\mathcal{L}(\theta) = \left[\left(\delta(t)\right)^2\right] \quad (10)$$

Donde $\delta(t) = R(s_t, a_t, s_{t+1}) + \gamma \cdot \theta^{Q'} - \theta^{Q}$ se conoce como TD-error. Una vez calculado el error, se procede con el entrenamiento de θ^{Q} que contiene los pesos que permiten la aproximación de los valores de $Q(s, a)$ [21]. Respecto al entrenamiento de θ^{Q}, se utiliza el método tradicional de aprendizaje de propagación hacia atrás.

Por otro lado, θ^{Q} y $\theta^{Q'}$ generalmente tienen la misma arquitectura, siendo $\theta^{Q'}$ una copia de θ^{Q} que se actualiza después de cada ω pasos, por lo anterior solo se entrena θ^{Q} [16, 21]:

$$\theta^{Q'} \leftarrow \theta^{Q} \quad (11)$$

El motivo para mantener $\theta^{Q'}$ estático por ω pasos es para evitar problemas de objetivos móviles que pueden afectar la convergencia de la red neuronal [21] . Considerando lo anterior, la arquitectura de implementación de DQN que normalmente se encuentra en la literatura se muestra en la Figura 3:

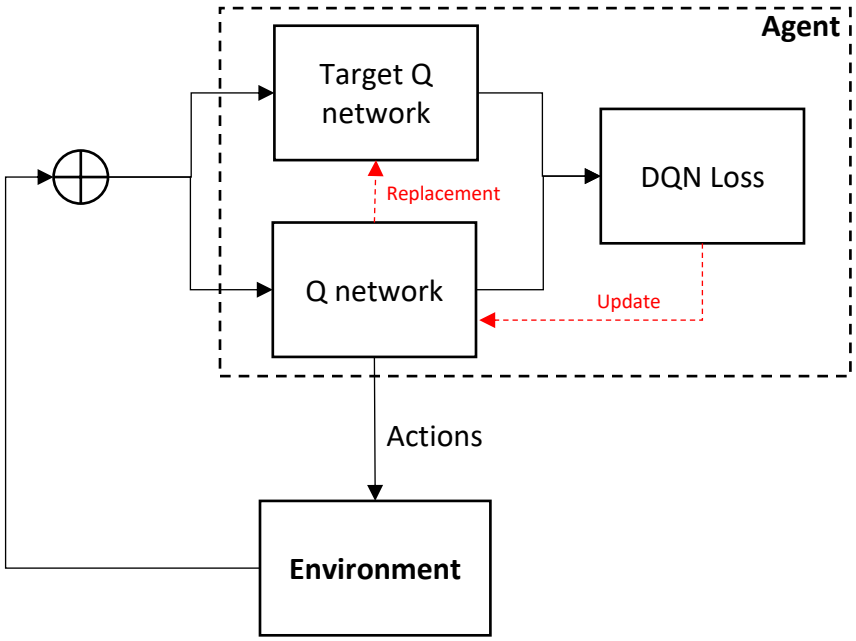


Figura 3. Arquitectura DQN implementada en el proyecto. Fuente: [21].

Finalmente, es importante recordar que el uso de redes neuronales como función de aproximación en Q-learning, permite el manejo de estados continuos, lo que es un avance

importante respecto a su versión clásica discreta. Sin embargo, existe una limitación con DQN en la siguiente expresión matemática:

$$argmaxQ(s', a'; \theta')$$

Esta expresión exige que se debe seleccionar la acción ***a'*** donde el valor Q sea máximo, es decir, este método está limitado a problemas donde las acciones son finitas para que sea posible ejecutar esta operación matemática [21].

2.1.7. Advance Actor Critic - A2C:

Los métodos basados en gradientes de política (*Policy Gradient* - PG) buscan aprender directamente la mejor política con una función parametrizada θ [23]:

$$\pi(a|s; \theta) \tag{12}$$

Los métodos PG tienen su fundamento teórico en el teorema de gradientes de política (*Policy Gradient Theorem*) donde se optimiza el retorno esperado $\mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta}[R(t)]$ a través del descenso de gradiente ($\nabla\mathcal{J}(\theta)$) de tal forma que se optimice el retorno esperado. Su expresión final es la siguiente [8]:

$$\nabla\mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta}[\nabla ln\pi(a|s, \theta) \cdot R(t)] \tag{13}$$

Sin embargo, los métodos PG tienen un inconveniente, y es que al igual que en el método de Montecarlo, se debe esperar al final de un episodio para calcular la recompensa acumulada en cada paso. Esto puede implicar que, si se obtienen altas recompensas, todas las acciones que fueron tomadas son buenas, aunque algunas hayan sido malas (ver figura 4).

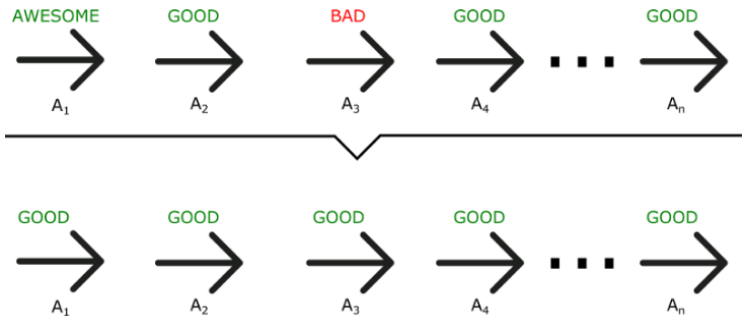


Figura 4. Pasos con su respectiva acción A_n y sus recompensas, seguida en dos trayectorias distintas. Fuente: Tomado de T. Simonini[1].

[1] https://www.freecodecamp.org/news/an-intro-to-advantage-actor-critic-methods-lets-play-sonic-the-hedgehog-86d6240171d/

De la Figura 4 se observa que, aunque A_3 fue una mala acción, todas las acciones en promedio fueron buenas porque lo que importa es la recompensa total. Esto implica que, para tener una política óptima que logre discriminar malas acciones, se necesitan muchas muestras, lo que hace que el aprendizaje sea lento porque toma mucho tiempo en converger. El esquema Actor-Critic presenta una mejor función de puntuación, porque en vez de esperar hasta el final del episodio para calcular *R(t)* de la ecuación (13), se puede hacer una actualización en cada paso como en *TD learning* a través de *Q(s,a)* [22]. Entonces, se puede reescribir la ecuación (13) como:

$$\nabla \mathcal{J}(\theta) = \mathbb{E}_{\pi_\theta}[\nabla ln\pi(a|s,\theta) \cdot \boldsymbol{Q}(\boldsymbol{s},\boldsymbol{a})] \tag{14}$$

De esta forma, se observa que el algoritmo de Actor-Critic consiste en dos modelos [23], donde se combinan los métodos basados en valor y en política:

- ***Actor***: Es un algoritmo PG que actualiza los parámetros de la política, en la dirección sugerida por el crítico. Es el encargado de seleccionar que acción tomar, y de estimar $\nabla ln\pi(a|s,\theta)$ de la ecuación (14).

- ***Critic***: Es un algoritmo basado en valor. Su objetivo es decirle al actor que tan buena fue la acción tomada y como debería ajustarla. Es el encargado de estimar $Q(s,a)$ de la ecuación (14).

Es importante mencionar respecto al Crítico que existen otras alternativas para estimar la función valor y, por tanto, generar variantes del algoritmo. Cuando se utiliza *Q(s,a)* como función valor, el algoritmo se conoce como Q Actor-Critic (QAC). Sin embargo, si se utiliza como función valor la función de ventaja (*advantage function*), cuya expresión se detalla en la ecuación (15) [22], se tiene como resultado la variación *Advantange Actor Critic* (A2C):

$$\mathrm{A}(s_t, a_t) = r_{t+1} + \gamma \cdot \mathrm{V}(s_{t+1}) - \mathrm{V}(s_t) \tag{15}$$

A2C presenta una mejora respecto a QAC en que es menos volátil, lo que facilita la convergencia y aprendizaje del algoritmo, especialmente si se utilizan redes neuronales para modelar el Actor y el Crítico en problemas de altas dimensiones. Un esquema general de alto nivel de este método se muestra en la Figura 5:

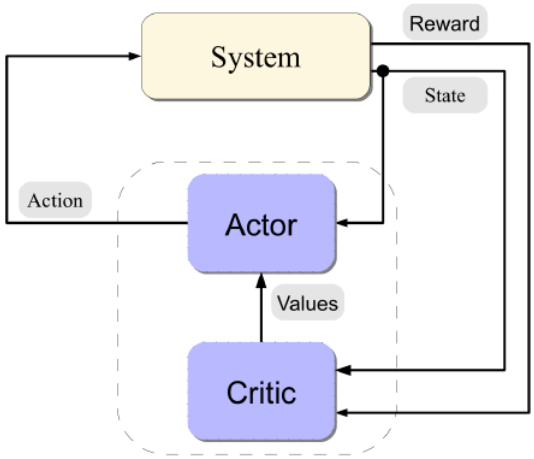


Figura 5. Arquitectura general del método Actor-Critic. Fuente: Tomado de Szepesvári[2]

Respecto al diseño de la red neuronal, es común encontrar en la literatura dos posibles arquitecturas para el actor (π_θ) y el crítico (V_ϕ^π), dependiendo si se comparten o no parámetros entre los modelos, tal como se ilustra en la Figura 6:

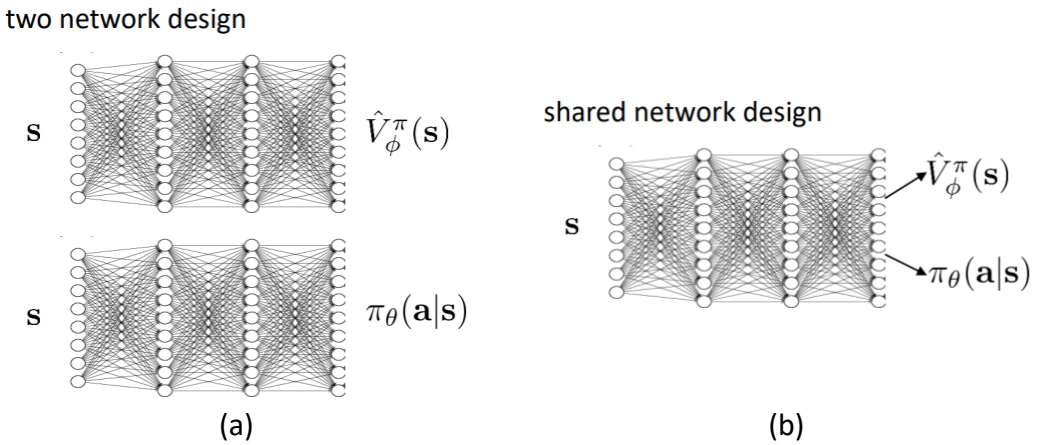


Figura 6. Posibles diseños de redes neuronales para el método Actor-Critic. La figura (a) representa el caso cuando la política (actor) y la función valor (crítico) se calculan a través de redes independientes. Por el contrario, la figura (b) representa un diseño donde se utiliza una única red neuronal para estimar tanto la política como la función valor. Fuente: Tomado de S. Levine[3].

La figura 6(a) representa el caso cuando la política (actor) y la función valor (crítico) se calculan a través de redes independientes. Por el contrario, la figura 6(b) representa un diseño donde se utiliza una única red neuronal para estimar tanto la política como la función valor. Otro aspecto importante para considerar en la Figura 6 es el cálculo de la función de pérdida de la red neuronal para cada caso. Para la figura (a) existen dos funciones de pérdida independientes, una asociada al Actor ($\mathcal{L}_{actor}$) y otro al Crítico ($\mathcal{L}_{critic}$). Por otra

[2] https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf
[3] http://rail.eecs.berkeley.edu/deeprlcourse-fa18/static/slides/lec-6.pdf

parte, para la figura 6(b) se calcula una pérdida global calculada como la suma de las pérdidas del Actor y el Crítico ($\mathcal{L} = \mathcal{L}_{actor} + \mathcal{L}_{critic}$).

La expresión $\mathcal{L}_{actor}$ se calcula normalmente como la función de entropía cruzada (cross-entropy) adaptada al contexto del algoritmo A2C:

$$\mathcal{L}_{actor} = -\sum_{1}^{n} \delta(t) \cdot log\left(\rho\left(V_{\phi}^{\pi}\right)_{a_i}\right) \tag{16}$$

Donde $\delta(t)$ es el TD-error y $\rho\left(V_{\phi}^{\pi}\right)_{a_i}$ cuantifica la salida de la red neuronal como una distribución de probabilidad de la acción. El signo de $\delta(t)$ sugiere que un valor positivo son tendencias que se deberían fortalecer en el futuro, mientras que valores negativos son tendencias que se deberían debilitar.

Finalmente, $\mathcal{L}_{critic}$ se calcula de igual manera a lo descrito para la función de pérdida del algoritmo DQN, que corresponde al cálculo de $\delta(t)^2$.

2.2. ESTADO DEL ARTE.

El problema de optimización de despacho energético hidrotérmico permite analizar el comportamiento del sistema en el mediano o largo plazo para analizar riesgos de desabastecimiento, precio de bolsa, necesidades de expansión, etc. Su modelado matemático está basado normalmente en el modelo de despacho de mínimo costo, considerando la restricción de balance hidrológica y de atención de la demanda. Para el caso colombiano, la base es el modelo de despacho ideal definido para el mercado de energía mayorista colombiano [24], y el modelo de análisis energético de largo plazo utilizado por XM[4] descrito en [5].

Este problema de despacho, y sus variaciones, son problemas ampliamente estudiados en la literatura, con diferentes líneas de investigación de abordaje. En [2, 25] describen modelos matemáticos para abordar la optimización estocástica, tales como: modelo equivalente determinístico y árboles de decisión. Sin embargo, se demuestra que estos enfoques no son viables dado que la complejidad del problema crece exponencialmente con el número de escenarios y variables de decisión. Por otra parte, Yuping, Panos y Qipeng [3] hacen un resumen metodológico de algoritmos disponibles para resolver el despacho de forma determinística.

[4] XM: Compañía de Expertos en Mercados S.A E.S.P, es una empresa del Grupo ISA especializada en la gestión de sistemas de tiempo real, la administración del mercado de energía mayorista y el desarrollo de soluciones y servicios de energía e información. Ejerce funciones de Centro Nacional de Despacho -CND-, Administrador del Sistema de Intercambios Comerciales -ASIC- y Liquidador de Cuentas de Cuentas de cargos por Uso de las redes del Sistema Interconectado Nacional - LAC en el sector eléctrico colombiano.

Un aporte muy relevante para el manejo estocástico de variables de decisión en el despacho fue planteado por Pereira y Pinto [5], donde desarrollan el algoritmo "*Stocastic Dual Dynamic Programming"* (SDDP) que consiste en utilizar programación dinámica para definir una política óptima de operación de las centrales hidroeléctricas mediante técnicas de descomposición. Este modelo es ampliamente utilizado en la actualidad, pero la complejidad del algoritmo [26], sus problemas de convergencia y su fundamento en un algoritmo de búsqueda que no garantiza la solución óptima global, hace que se sigan explorando nuevas estrategias.

En [27], Charles, Ansari y Khalid transforman el problema estocástico en un modelo determinístico equivalente de despacho para diferentes distribuciones de probabilidad, pero no disminuye la complejidad computacional del problema. En [6, 28, 29] utilizan algoritmos metaheurísticos para aproximar la solución, pero sin garantía de optimalidad. También hay enfoques más avanzados, como [7, 30, 31, 32], donde cambian la función objetivo de mínimo costo por un problema de competencia de mercado, con la finalidad de simular el interés económico de los agentes y generar un despacho de las centrales más aproximado a la realidad. Sin embargo, parten del supuesto de información perfecta.

Otro enfoque que se ha venido utilizando recientemente consiste en la formulación del despacho económico como un MDP [8] para la aplicación de algoritmos de RL. Entre las aplicaciones de RL en el campo del sector eléctrico se tiene las propuestas de [33, 34], donde proponen un algoritmo de Q-learning para solucionar el despacho considerando las pérdidas por transmisión. No consideran la caracterización hidrológica, muy importante en muchos contextos como para el caso colombiano. En [35], Imthias y Jasmine consideran la aleatoriedad de los costos de producción, pero enfocado en centrales térmicas. Abouheaf, Haesaert, Lee y Lewis [36] emplean Q-learning para solucionar el problema de no convexidad del despacho económico. Sin embargo, no considera la evolución en el tiempo de las centrales. Ali, Guerci y Cincotti, en [37] presentan una aplicación de RL para el sistema eléctrico italiano, pero sin considerar la incertidumbre en la hidrología.

Aunque las investigaciones más recientes siguen enfocadas en abordar la no linealidad del modelo matemático mediante RL [38], o la aplicación de RL en temas de actualidad como las micro-redes [39], también aparecen trabajos recientes relacionados con el problema de estudio, como Remya, Johnson y Ahamed [40] que analizan un problema similar pero considerando un espacio de estados y acciones discretos aplicando la versión discreta de Q-learning, lo que representa una fuerte limitación en aplicaciones prácticas por problemas de dimensionalidad. También, Yu y Li [19] resuelven el problema de despacho a largo plazo para centrales térmicas considerando la incertidumbre en la demanda, proponiendo un algoritmo de RL profundo para solucionar el problema de alta dimensionalidad. De los trabajos se observa que es necesario utilizar técnicas avanzadas de RL para abordar al gran número de estados del problema. Sallans y Hinton en [9] describen un caso de éxito en la aplicación de RL profundo en un problema de alta dimensionalidad.

Como aplicación de RL para el sector eléctrico colombiano, se destaca [38], donde Gallego, Duarte y Delgadillo definen un enfoque de aprendizaje multi-agente para la predicción de precios de oferta de las centrales a partir de los precios del mercado y la percepción de riesgo. Aunque la propuesta no considera el efecto temporal de las centrales ni la hidrología, se observa que al definir funciones de aprendizaje por planta es posible disminuir el espacio de exploración, pero con el costo de incluir estrategias de aprendizaje multi-agente que pueden hacer más complejo la modelación. Sobre esto último, dado que para el caso particular de análisis se supone un objetivo común de minimización costos, es posible simular la cooperación entre agentes mediante la maximización del beneficio de cada agente individualmente con experiencia compartida [39].

Considerando las investigaciones mencionadas, se observa que hay pocos desarrollos en el modelado del problema de despacho energético hidrotérmico incorporando conjuntamente la incertidumbre en las variables y la toma de decisiones secuenciales. En ese sentido, los resultados del presente proyecto aportan al estado del arte, proponiendo un enfoque que incorpora la incertidumbre de las variables en el despacho energético con la búsqueda de una política óptima que haga el mejor uso de los recursos disponibles. Dicho enfoque es basado en el paradigma de RL profundo de la inteligencia artificial.

3.1. DEFINICIÓN DEL MODELO MATEMÁTICO DE OPTIMIZACIÓN:

Para el diseño del modelo matemático de despacho económico que permita simular el comportamiento del sistema, se toma como base el modelo de despacho ideal definido para el mercado de energía mayorista colombiano [24] y el modelo de análisis energético de largo plazo utilizado por XM que corresponde al modelo descrito en [5]. Dichos modelos se usan para definir un modelo ajustado a la información pública disponible, pero que logre capturar en lo posible las reglas de operación del mercado de energía colombiano. El modelo matemático clásico de despacho a mínimo costo [2, 24] ajustado al alcance de la investigación es el siguiente, que permite minimizar los costos de producción de energía eléctrica es:

$$\min \mathrm{Z} = \sum_{t \in T} \sum_{p \in P} \left(C_p \cdot G_{p,t} \cdot FP_p \right) + \sum_{t \in T} \sum_{h \in H} \left(C_h \cdot D_{h,t} \cdot FC_h + 50 \cdot C_h Vert_{h,t} + 100 \cdot C_h \cdot EmbH_{h,t} \right) \quad (17)$$

Donde los índices representan:

P:	*Conjunto de centrales no hidroeléctricas.*
H:	*Conjunto de centrales hidroeléctricas.*
T:	*Conjunto de etapas o periodos de tiempo.*

Los *parámetros* son:

C_h, C_p:	*Costo de generación de la planta h o p.*
FP_p:	*Factor de planta de la central p.*
FC_h:	*Factor de conversión para convertir caudal en energía de la central h.*

Y las *variables de Decisión* son**:**

$G_{p,t}$:	*Generación de la planta p en el periodo t.*
$D_{h,t}$:	*Descarga de agua turbinada de la planta h en el periodo t.*
$Vert_{h,t}$:	*Caudal de agua vertida de la planta h en el periodo t.*
$EmbH_{h,t}$:	*Volumen de agua de holgura de la planta h en el periodo t.*

Sujeto a las Restricciones:

Balance de demanda verifica que la generación de las plantas sea igual a la demanda de energía eléctrica:

$$\sum_{p \in P} G_{p,t} \cdot FP_p + \sum_{h \in H} D_{h,t} \cdot FC_h = Dem_t, \quad \forall\, t \in T \tag{18}$$

Donde Dem_t es la Demanda de energía en el periodo *t*.

Los *límites de producción de energía* de las centrales térmicas son:

$$LG_p \leq G_{p,t} \leq UG_p, \quad \forall p \in P, \quad \forall t \in T \tag{19}$$

Donde, $\mathrm{UG_p}$ (upper generation) es la generación máxima y $\mathrm{LG_p}$ (lower generation) es la generación mínima de la planta p.

Los *límites de descarga de agua* de las centrales hidroeléctricas son:

$$LD_h \leq D_{h,t} \leq UD_h, \quad \forall h \in H, \quad \forall t \in T \tag{20}$$

Donde, $\mathrm{UD_h}$, es la descarga máxima y $\mathrm{LD_h}$ es la descarga mínima de la central h.

Los *límites de almacenamiento de agua en los embalses* introducen dos nuevas variables de decisión asociado al embalse en el periodo *t* y una variable de holgura para asegurar factibilidad del problema de optimización:

$$LE_{h,t} + EmbH_{h,t} \leq Emb_{h,t} \leq UE_h, \quad \forall h \in H, \quad \forall t \in T \tag{21}$$

Donde, $\mathrm{Emb_{h,t}}$ es el volumen de agua de la planta *h* en el periodo *t*. $\mathrm{UE_h}$ y $\mathrm{LE_p}$ hacen referencia a los límites de embalse máximo y mínimo, respectivamente, de la central *h.*

Los *Balances de masa* en los embalses se definen como:

$$Emb_{h,t-1} + Q_{h,t} \cdot FCV_h - D_{h,t} - Vert_{h,t} = Emb_{h,t}, \quad \forall h \in H, \forall t \in T \tag{22}$$

Donde, $\mathrm{Q_{h,t}}$ hace referencia al caudal de entrada en la planta *h* en el periodo *t.* $\mathrm{FCV_h}$ es un factor de conversión para convertir caudal de agua en volumen para la planta *h.*

Finalmente se define las condiciones de *no negatividad*:

$$\mathrm{Emb_{h,t}},\ \mathrm{D_{h,t}},\ \mathrm{G_{p,t}},\ \mathrm{Vert_{h,t}},\ \mathrm{EmbH_{p,t}} \geq 0 \tag{23}$$

Por simplicidad, se limita el modelo matemático de optimización mediante las siguientes suposiciones:

- No se consideran costos de arranque y parada de las centrales térmicas. Esto significa que en la función objetivo no se considerar costos fijos asociados a variables binarias que simulan el arranque de las centrales térmicas despachadas.
- Se manejan variables de holgura para el vertimiento ($Vert_{h,t}$) y el embalse mínimo ($EmbH_{h,t}$) para evitar problemas de factibilidad en la solución del modelo de optimización asociados al cumplimiento de la ecuación (21) y (22).
- Se supone precio de oferta de las centrales eléctricas igual a los otros costos variables (OCV[5]) pero suponiendo un desempate predefinido (C_h). De esta forma, se elimina problemas de inestabilidad en el modelo matemático.
- Los precios de oferta (C_h, C_p) y los factores de planta para las centrales térmicas (FP_p) y las centrales no despachadas centralmente se consideran constantes en todo el horizonte de optimización. Los precios y los factores son calculados con información histórica.
- No se considera la entrada de nuevas centrales al sistema de generación.

Para el caso de estudio colombiano, se consideran 28 centrales hidroeléctricas, 35 centrales térmicas y 148 centrales menores. Los datos usados en este trabajo para la formulación del modelo matemático se detallan en el ANEXO I "Descripción y Adquisición de los Datos". Suponiendo un horizonte de optimización de 24 meses, el modelo tendría un total de 5439 variables de decisión y 7807 restricciones, lo que exige el uso de un software de optimización para resolver el despacho energético. Los softwares que se utilizan en el proyecto son *Coin Or Branch and Cut* (CBC[6]) y *GNU Linear Programming Kit* (GLPK[7]) con licencia libre.

Para validar los resultados obtenidos en el proceso de optimización, se propone analizar el precio de bolsa[8], el cuál es un dato de consulta pública suministrado por XM (ver en tabla 4, precio de bolsa). El precio de bolsa se calcula como la oferta (C_h, C_p) de la última central que generó en el despacho económico ($G_{p,t}$ y $D_{h,t}$), ordenadas de menor a mayor costo, donde las centrales despachadas minimizan los costos de producción de la función objetivo[9] de la ecuación (17). De esta forma, es posible comparar de forma rápida la consistencia del despacho del modelo matemático con información real del sistema colombiano. Para esto, se alimenta el modelo con información perfecta de: demanda real de energía (Dem_t, ver

[5] Los OCV son los costos mínimos de la energía, los cuales son recaudados por el generador, pero no representan utilidad alguna.
[6] https://www.coin-or.org/
[7] https://www.gnu.org/software/glpk/
[8] Se puede consultar la definición de precio de bolsa para el contexto colombiano en: https://www.xm.com.co/Paginas/Mercado-de-energia/precio-de-bolsa-y-escasez.aspx
[9] Es decir, que el precio de bolsa se calcula una vez finaliza el proceso de optimización del modelo matemático.

en tabla 4, demanda de energía), embalses en cada mes ($Emb_{h,t}$, ver en tabla 4, reserva mensual de los embalses), y los caudales reales de los ríos ($Q_{h,t}$, ver en tabla 4, aportes mensuales de caudales de agua). No se considera la generación real de las centrales térmicas porque estás atienden la demanda restante que no pudo ser atendida por las centrales hidroeléctricas:

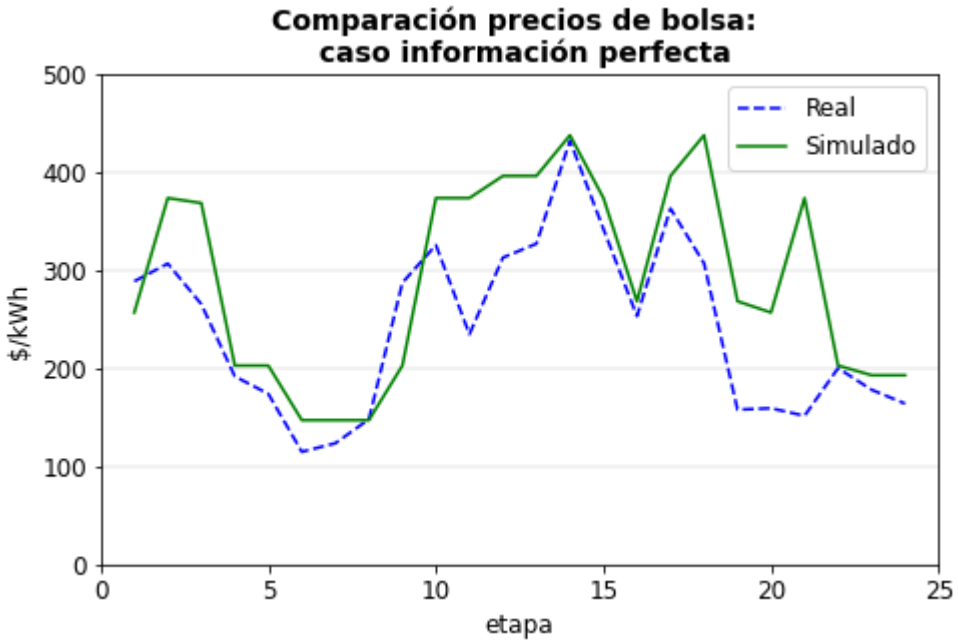


Figura 7. Comparación ente el precio de bolsa real 2019-2020 frente al precio simulado calculado por el modelo suponiendo información perfecta en la demanda, gestión de embalses y caudales de los ríos. Fuente: Elaboración propia

De acuerdo con la Figura 7, el modelo (línea continua) refleja en buena medida la dinámica real del sistema (línea discontinua), a pesar de los supuestos mencionados. Sin embargo, si se considera desconocido el nivel de los embalses, y se cambia por su mínimo histórico de cada mes, se puede observar en la Figura 8 un comportamiento diferente a la Figura 7:

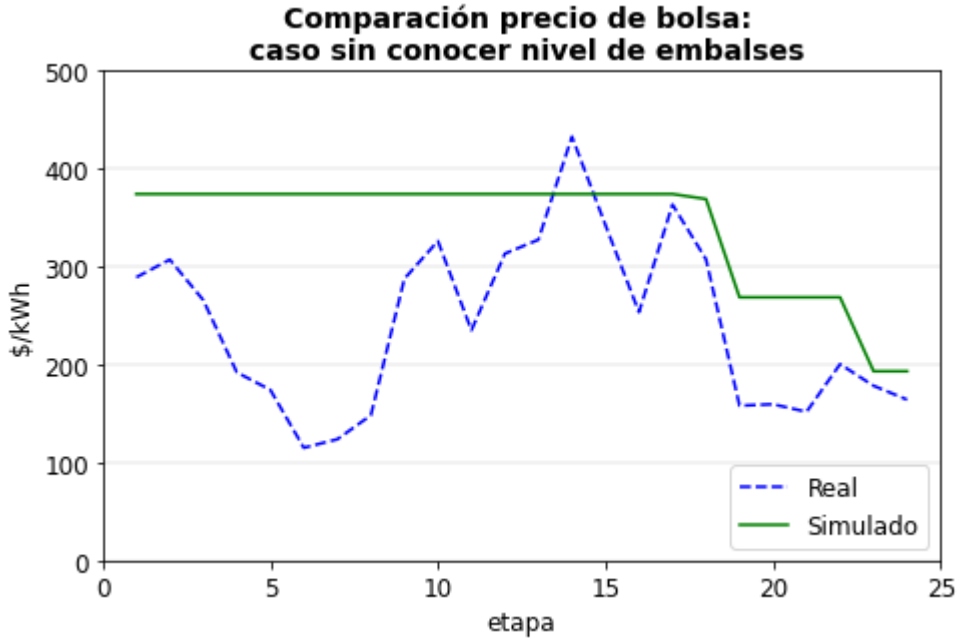


Figura 8. Comparación ente el precio de bolsa real 2019-2020 frente al precio simulado calculado por el modelo suponiendo embalses mínimos históricos e información perfecta en caudales de los ríos. Fuente: Elaboración propia

La Figura 8 refleja un comportamiento esperado por parte del modelo matemático (línea continua), donde se observa una caída del precio de bolsa en etapas finales, porque la función objetivo busca utilizar todo el recurso hídrico ($Emb_{h,t}$) posible. Lo anterior representa un problema, dado que demuestra la incapacidad del modelo matemático en representar la dinámica del sistema colombiano con una señal de precios plana, además del uso intensivo de los recursos hídricos en etapas finales que no refleja la incertidumbre en la demanda y los aportes de agua. Por ello, se requieren de alternativas como las descritas en el estado del arte o la presente propuesta.

3.2. MODELACIÓN DE VARIABLES CON INCERTIDUMBRE.

Parte de los grandes retos en la solución del problema del despacho energético es la modelación de la incertidumbre en los valores de variables como la demanda de energía y los aportes de agua de los ríos. Para lo anterior, y aprovechando el marco metodológico del aprendizaje reforzado (RL), se modela los parámetros Dem_t y $Q_{h,t}$ como parámetros aleatorios del entorno, que cambian en cada episodio en el proceso de aprendizaje del algoritmo.

3.2.1. Incertidumbre en la demanda:

Para modelar la demanda, se propone seguir con la filosofía de la UPME donde se generan 3 escenarios de demanda: bajo, medio y alto. Se estima la demanda promedio del 2019 y

2020 considerando un crecimiento anual del 2.5%, 3.5% y 4.5%, tomando como base el año 2018. Una vez se selecciona el escenario de demanda, este se afecta por un valor aleatorio con distribución normal de media 1 y desviación estándar de 0.015 para simular un error en el pronóstico. El valor aleatorio se techa para que no sea inferior a 0.985. La Figura 9 representa los posibles rangos de valores de demanda que serán utilizados en el escenario medio de demanda:

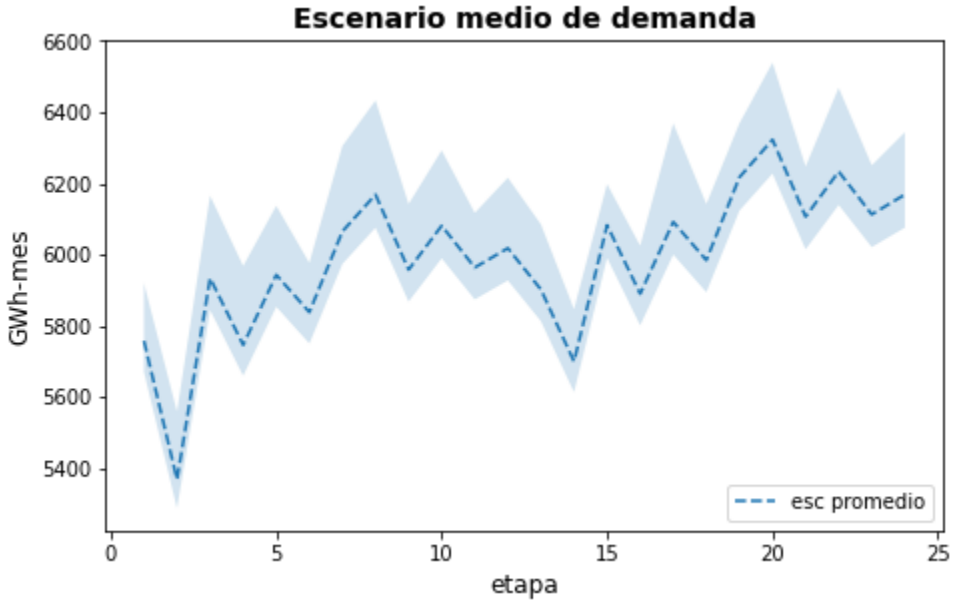


Figura 9. Escenario medio de demanda con bandas de variación por aleatorio normal. Fuente: Elaboración propia

3.2.2. Incertidumbre en los caudales:

Para modelar los aportes de agua de los ríos, se propone generar 5 escenarios climáticos tomando análogos históricos: húmedo (año 2000), seco (año 2015), neutro (año 2013), seco a neutro (año 2003) y seco a húmedo (año 2007). Una vez se selecciona el escenario climático, este se afecta por un valor aleatorio con distribución normal de media 1 y desviación estándar de 0.05 para simular un error en el pronóstico. La Figura 10 representa los posibles rangos de valores de caudales que serán utilizados en un escenario neutro para el río Nare:

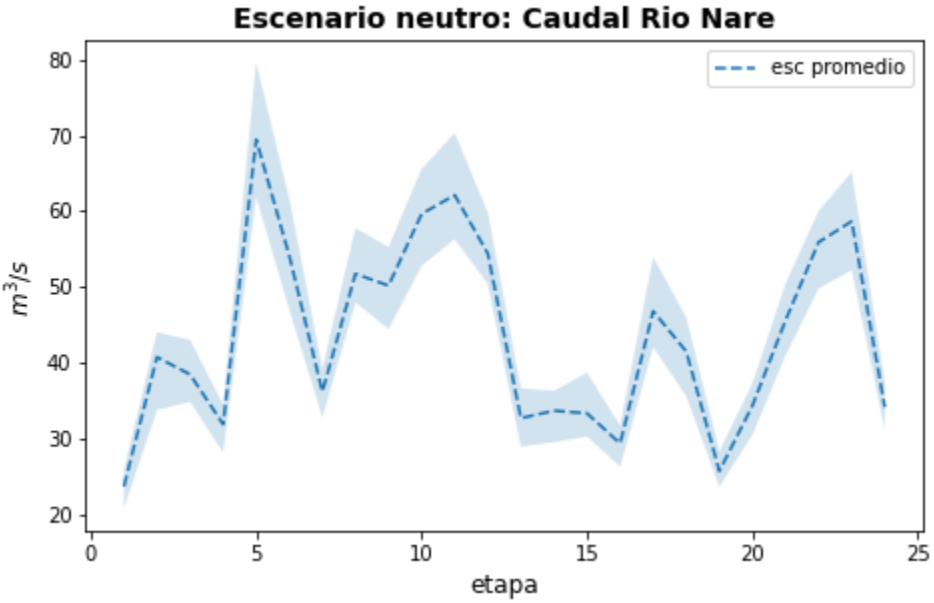


Figura 10. Escenario neutro de caudales para el río Nare con bandas de variación por aleatorio normal. Fuente: Elaboración propia

CAPÍTULO 4: FORMALIZACIÓN DEL PROBLEMA COMO UN PROCESO DE DECISIÓN DE MARKOV

En el capítulo 3 se describe la dificultad que tiene el modelo de despacho energético de mínimo costo para representar la dinámica del caso de estudio colombiano, cuando no utiliza información perfecta. Para abordar esta limitación, el presente proyecto propone combinar el modelo matemático de optimización con el paradigma de RL, transformando el problema como un proceso de decisión de Markov, de tal manera que se pueda estimar una política robusta que considere la variabilidad en la demanda y los aportes de agua. A continuación, se formaliza el problema de decisión de Markov aplicado al contexto del despacho energético de mínimo costo, describiendo al agente, las acciones, el entorno, y el modelado de la incertidumbre. Al final, se indican las extensiones al algoritmo de RL profundo usadas.

4.1. AGENTE (AGENT).

El agente tiene a cargo el aprendizaje, la toma de decisiones y la interacción con el entorno. Sin embargo, existe poco desarrollo en casos cuando un agente debe tomar simultáneamente varias acciones, como es el caso del presente proyecto, donde el agente debe determinar los niveles de descarga para cada central hidroeléctrica al mismo tiempo. La interacción entre el agente y el entorno para un caso de múltiples acciones [39, 40, 41], se puede modelar bajo el esquema mostrado en la Figura 11.

En el esquema, el **agente** toma una **acción** $\boldsymbol{A_N}$ para cada central hidroeléctrica basado en el estado actual conjunto **(*join state*)** determinado por el nivel de los embalses, aporte de agua en cada central, y nivel de demanda. Las acciones se agrupan como una única acción (***join action***) que afecta al **entorno**. El entorno procesa la acción conjunta, y limita la generación máxima de las centrales hidroeléctricas para proceder con la optimización del despacho energético. Una vez finalizada la minimización se llega a un **nuevo estado** (*Next State*), y el valor de la función objetivo se entrega como **recompensa**. A partir del estado anterior, el nuevo estado y la recompensa, el agente procede con el entrenamiento, y se repite el ciclo hasta converger.

El diseño del agente (***Agente DRL***) se realiza a través de algoritmos de RL. El proyecto propone experimentar con los algoritmos DQN y Actor-Critic descritos previamente en el capítulo 2.

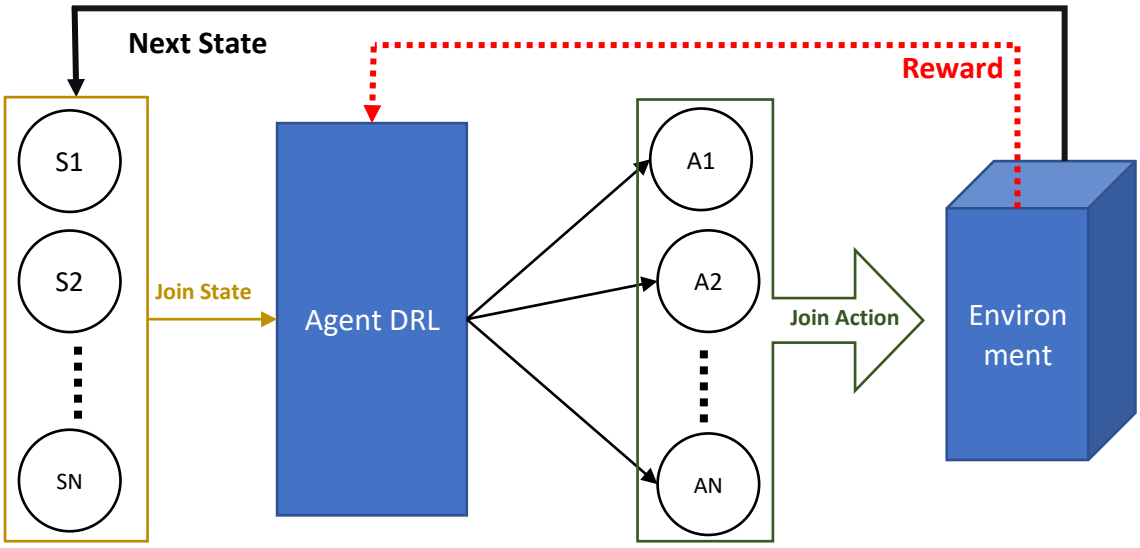


Figura 11. Esquema general para aplicación de RL para el caso de un agente con múltiples acciones. Fuente: Elaboración propia

4.2. ACCIONES (ACTION).

La modelación del despacho energético descrito previamente supone que las centrales térmicas cuentan con recursos ilimitados, lo que significa que no importa el punto en el tiempo, siempre es posible contar con sus aportes en generación. Por el contrario, las centrales hidroeléctricas dependen de los aportes de agua $Q_{h,t}$ y sus reservas en el embalse ($Emb_{h,t}$), lo que significa que las decisiones están acopladas en el tiempo. De acuerdo con lo anterior, solo es relevante simular las acciones que debe tomar el agente para las centrales hidroeléctricas, porque una decisión en el uso de los embalses en *t* puede tener impacto en *t+1*.

Se propone simular la acción como un número entre cero y uno asignado para cada central de manera independiente. Por ejemplo: si la acción es 1 entonces la central puede usar el 100% de su capacidad máxima de descarga, pero si es 0.2 solo puede usar el 20% de su capacidad máxima. Considerando los algoritmos de RL a analizar en la propuesta, es necesario definir la acción para el caso continuo y discreto:

- Para el caso continuo, el dominio de acciones válido que se puede elegir para cada central hidroeléctrica *h* es:

$$Dominio: \{A_h \in R / 0 \leq A_h \leq 1\}, \quad \forall h \in H \tag{24}$$

- Para el caso discreto, el conjunto de acciones válido que se puede elegir para cada central hidroeléctrica *h* es:

$$A_h = \{0\%, 10\%, 20\%, 30\% \dots, 100\%\}, \quad \forall h \in H \tag{25}$$

Finalmente, se define la acción conjunta, continua o discreta, como la unión de todas las acciones tomadas por cada central:

$$A_{join} = \bigcup_{h \in H} A_h \tag{26}$$

4.3. ENTORNO (ENVIRONMENT).

El entorno es el punto de encuentro entre el modelo matemático de despacho energético y el aprendizaje reforzado. Para su integración, es necesario hacer modificaciones al modelo matemático, de tal manera que sea posible definir la interacción de las entradas (acción) y salidas (estado y recompensas) descritas en la Figura 1. A continuación, se describen esas modificaciones.

4.3.1. Estado (State):

Al igual que la Acción, el Estado se define a partir de parámetros asociados a las centrales hidroeléctricas por su dependencia de las decisiones en el tiempo. Se define el estado como la tupla: nivel de embalse de cada central que representa los recursos disponibles ($Emb_{h,t}$), nivel de aportes de agua que representa los nuevos recursos ($Q_{h,t}$), y nivel de demanda eléctrica que representa el requerimiento de energía a suministrar (Dem_t). Así, el Estado se representa a través de las siguientes expresiones:

El *nivel de embalse para cada central h* en el mes *t* se calcula como:

$$\%Emb_{h,t} = \frac{Emb_{h,t}}{UE_h}, \quad \forall h \in H \tag{27}$$

Donde:

t: valor entero que representa es el mes actual*, %Emb:* valor continuo que representa el porcentaje de agua almacenada en el embalse *h* en el mes *t, $Emb_{h,t}$*, embalse actual de la central *h* en el periodo *t* en unidades de volumen**,** *UE_h*: Embalse máximo de la central *h* en unidades de volumen.

Por otra parte, el *nivel de aportes de agua del sistema* en el mes *t* se calcula como:

$$\%Aportes_t = \frac{\sum_{h \in H} Q_{h,t}}{UQ_h} \tag{28}$$

Donde, *%Aportes:* Valor continuo que representa la entrada de caudal de agua en todas las centrales en el mes *t*, $Q_{h,t}$*:* Caudal de entrada de agua en la central *h* en el periodo *t* en unidades de caudal, UQ_h: Caudal de entrada máximo histórico de agua en la central *h* en unidades de volumen.

Finalmente, el *nivel de demanda del sistema* en el mes t se calcula como:

$$\%Dem_t = \frac{Dem_t}{UDem} \tag{29}$$

Donde, *%Dem:* Valor continuo que representa el nivel de demanda en el mes *t*, Dem_t*:* Demanda de energía en el periodo *t*, *UDem:* Demanda máxima histórico de agua en la central *h* en unidades de energía.

De esta forma, el estado se compone por la unión conjunta de las expresiones anteriores, tomando la siguiente forma:

$$S_t = \left(\bigcup_{h \in H} \%Emb_{h,t} \, , \%Aportes_t, \%Dem_t \right) \tag{30}$$

4.3.2. Recompensa (*Reward*).

La recompensa es una salida del entorno. Para el contexto del proyecto se define como el negativo de la función objetivo de la ecuación (17), de esta forma el agente buscará maximizar su recompensa buscando la política que minimice la función objetivo. Sin embargo, es importante considerar los siguientes puntos:

- Una acción puede llevar a situaciones operativas indeseables tales como: vertimiento de embalses o incumplimiento de niveles mínimos operativos. Por lo anterior cada agente tendrá una penalización cuando se presenta dicha situación.
- Como se ha mencionado, parte de las desventajas del modelo matemático tradicional radica en el uso de todo el recurso de agua posible. Por lo anterior, se propone incluir una recompensa positiva que incentive mantener niveles altos de embalses.

De acuerdo con lo anterior, la recompensa para un paso del algoritmo que consiste en pasar del tiempo *t* a *t+1* es:

$$R_h = -C_1 \cdot FO - C_2 \cdot V_h - C_3 \cdot EH_h + C_4 \cdot \%Emb_h, \quad \forall h \in H \tag{32}$$

Donde V_h y EH_h son variables que cuantifican el uso de las variables de holgura de vertimiento y embalse, respectivamente, de la central *h*. La expresión *FO* hace referencia a la función objetivo de la ecuación (17), para atender la demanda en el tiempo t. $\%Emb_h$ es

el nivel del embalse de la central *h*. Finalmente, los parámetros C_1, C_2, C_3 y C_4 son factores de escala que buscan que los órdenes de magnitud de cada componente sean cercanos a uno, con dos objetivos: hacer comparables los componentes de la recompensa y facilitar el aprendizaje de las redes neuronales.

Como la recompensa se calcula para cada central *h*, se calcula la recompensa general del paso *t* al *t+1* (R_t) como la recompensa promedio obtenida de cada central. El agente avanza hasta alcanzar el horizonte de optimización *T*. Como resultado se obtienen *T* recompensas para un episodio. De esta forma el retorno esperado $R_{episodio}$ se calcula de la siguiente forma:

$$R_{episodio} = R_1 + R_2 + \cdots + R_T \tag{33}$$

Donde

$$R_T = \frac{\sum_{h \in H} R_h}{H} \tag{34}$$

H: Número de centrales hidroeléctricas.
$R_{episodio}$*:* Retorno esperado del episodio.
R_h: Recompensa de la central *h.*

4.3.3. Interacción del modelo matemático con la acción del agente:

La acción es una entrada para el entorno, por lo tanto, es necesario definir como esta entrada afecta al modelo matemático de optimización. Para lo anterior, es necesario modificar la restricción asociada a los límites de producción descritos en la ecuación (20) multiplicando el parámetro de descarga máxima (*UD*) por un nuevo parámetro A_h[10], de la siguiente forma:

$$\mathrm{L}D_h \leq D_{h,t} \leq UD_h * A_h, \quad \forall h \in H \tag{31}$$

Donde A_h es la acción tomada por el agente para la central *h*. De esta forma, se limita que la variable de decisión $D_{h,t}$ no supere el tope $UD_h * A_h$, que corresponde a la limitación en descarga que se busca simular. La ventaja de este diseño es que es válido para acciones continuas y discretas.

[10] Como el parámetro LQ se supone cero para esta investigación, es válido aplicar directamente el parámetro A_h al caudal máximo. De lo contrario sería necesario modificar la descarga máxima como: $(UD_h - LD_h) * A_h + LD_h$

4.3.4. Modelación de la Incertidumbre a través del entorno:

En la sección 3.2 se definió como modelar la incertidumbre en las variables de demanda y de caudales que caracterizan al entorno, de tal manera que el entorno no sea determinista, sino estocástico. Por ejemplo: cada vez que el agente esté en el mes 1, aunque la acción del agente siempre sea igual, el estado siguiente y la recompensa serán diferentes porque el entorno en cada ocasión selecciona aleatoriamente un escenario actual climático y otro de demanda. La interacción del agente con este entorno entrega como resultado una política de operación, cuyas decisiones tienen en cuenta la incertidumbre de la demanda y los aportes de agua (determinado por el escenario climático).

4.4. EXPLORACIÓN - EXPLOTACIÓN.

La exploración y explotación es un aspecto importante en el diseño de un algoritmo de RL porque es el elemento encargado de definir cuándo se puede explorar el espacio de estados, lo cual quizás puede conllevar a mejores recompensas, y cuándo explotar los mejores estados encontrados. Para el desarrollo de este proyecto se utilizan dos esquemas diferentes de exploración-explotación:

4.4.1. Epsilon-Greedy:

El primer esquema y el más clásico en RL es épsilon-greedy, donde con una probabilidad de ϵ se explora, mientras que con una probabilidad de $(1-\epsilon)$ se explota (ver algoritmo en tabla 1, pasos 5 al 8). Sin embargo, la exploración es importante en etapas tempranas del entrenamiento [16], motivo por el cual se elige la variación de épsilon-greedy con decaimiento para lograr más exploración en etapas tempranas y más explotación en etapas finales. Para lo anterior, se inicializa un nuevo parámetro $\in\_$ en uno, que disminuye linealmente en cada llamada al algoritmo restando un valor de decaimiento (paso 3 de la tabla 1); luego, se elige el valor máximo entre $\in$ y $\in\_$ (paso 4 de la tabla 1), para finalmente aplicar el algoritmo clásico de épsilon-greedy. Este esquema es apropiado para el método de DQN.

Tabla 1. Algoritmo $\in$-greedy para la elección del exploración o explotación del agente.

Algoritmo 1 *Algoritmo $\in$-greedy para seleccionar exploración-explotación*
Entrada: *valor aleatorio, valor de decaimiento.*
Parámetros: *parámetro de exploración $\in$.*
Salida: *decisión si explora o explota.*
1: Initialize p with zero, $\in\_$ with one
2: p = rand(0,1)
3: $\in\_$ = $\in\_$ - decay
4. $\in$e = max($\in\_$, $\in$)
*5. **if** p < $\in$e:*

6. *explore*
7. ***else:***
8. *exploit*

4.4.2. Exploración mediante distribución Gaussiana:

Para los métodos que utilizan acciones continuas, se utiliza el método de exploración gaussiana, que consiste en seleccionar una acción aleatoriamente siguiendo una distribución normal de probabilidad. Para el caso de los algoritmos Actor-Critic, la red neuronal del Actor se diseña de tal manera que ella selecciona una acción aleatoriamente mediante dos salidas que representan la media y la varianza. A medida que el Actor aprenda a tomar mejores acciones, la media converge a un valor óptimo y la varianza disminuye, haciendo que se genere el efecto de mayor exploración en etapas iniciales y mayor explotación en etapas finales.

4.5. AJUSTES DE LOS ALGORITMOS DE APRENDIZAJE REFORZADO.

En esta sección se presentan las mejoras propuestas en los algoritmos usados en este proyecto.

4.5.1. DQN:

El algoritmo DQN desarrollado en el proyecto se describe en el ANEXO II (ver tabla 6). Los hiper parámetros se describen en la tabla 5. Para mejorar la estabilidad en la convergencia del algoritmo se reemplaza la función de pérdida definida en la ecuación (9) por la función de Huber definida en la ecuación (35) (ver paso 21 en la tabla 6). Lo anterior se propone para reducir la volatilidad en el cálculo del error, lo que favorece el entrenamiento de la red neuronal. La función de Huber se comporta como una función de pérdida cuadrática para valores pequeños de entrada, y como una función de pérdida lineal para valores grandes de entrada. En el caso particular de DQN, la función de pérdida de Huber se calcula como:

$$\mathcal{L}(\theta^Q) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[\begin{cases} 1/2\left(\delta(t)\right)^2, & para \;\; \delta(t) \leq 1 \\ (|\delta(t)| - 1/2), & en\ caso\ contrario \end{cases}\right] \qquad (35)$$

La Figura 12 muestra visualmente la diferencia entre varias funciones de pérdida, donde se observa que la función de Huber es más tenue.

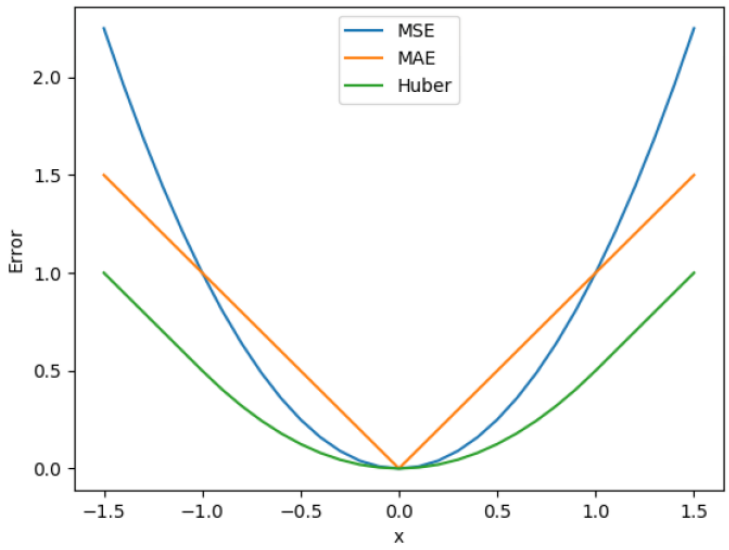


Figura 12. Gráfica de varias funciones de pérdida. Fuente: [42]

La expresión ***U(D)*** de la ecuación (35) hace referencia al mecanismo ***Experience Replay*** [21, 43], que consiste en una memoria D donde se almacenan todos los pasos ejecutados por el agente en cada episodio $\boldsymbol{e_t = (S_t, A_t, R_t, S_{t+1})}$ (ver paso 13 en la tabla 6). Luego, cuando se tengan suficientes datos, se toma una muestra aleatoria uniformemente distribuida de D (ver paso 15 en la tabla 6), para proceder con el entrenamiento de la red *Q-network* (ver paso 21 en la tabla 6). Este mecanismo se recomienda ampliamente en la práctica porque soluciona problemas de correlación entre los datos que afectan el aprendizaje de las redes neuronales.

4.5.2. A2C:

El algoritmo A2C desarrollado en el proyecto se describe en el ANEXO II (ver tablas 9 y 10). Los hiper-parámetros se describen en la tabla 8. De acuerdo con la descripción del algoritmo, la actualización de la política se realiza por cada paso del agente (S_t, A_t, R_t, S_{t+1}) (ver pasos 11 al 15 de la tabla 9), donde es posible que se pueda generar problemas en el entrenamiento de la red neuronal al no generalizar de manera apropiada usando un dato al tiempo. Como alternativa, se encuentran desarrollos de computación paralela donde se ejecuta el mismo estado tomando una muestra de acciones, y con esta muestra de posibles (S_t, A_t, R_t, S_{t+1}) se procede con el entrenamiento. Sin embargo, para el contexto del proyecto se considera otra variación (ver algoritmo en tabla 10), usando como muestra todos los pasos tomados por el agente en el horizonte de optimización T (ver pasos 10 al 13 de la tabla 10), debido a que T es un número fijo (lo que implica que el agente siempre ejecuta el mismo número de pasos) de baja magnitud.

CAPÍTULO 5: EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

Este capítulo presenta un análisis experimental de los diferentes agentes propuestos en este trabajo para el problema bajo estudio. En particular, se presenta el caso de estudio considerado, las métricas de calidad para evaluar los agentes, para continuar con la descripción del uso de los diferentes agentes (aleatorio, DQN, y A2C) en el caso de estudio, y una comparación entre ellos. Finalmente, se analiza detalladamente el comportamiento de la política del mejor agente en el caso Colombiano.

5.1. DESCRIPCIÓN DEL CASO DE ESTUDIO.

El caso de estudio consiste en simular el sistema de generación colombiano entre los periodos 2019-2020 (horizonte de optimización de 24 meses). Este periodo tiene la particularidad de ser cálido en el 2019 hasta principios del 2020 (lo que representa bajos aportes de agua), y un periodo alto de lluvias a finales del 2020.

Inicialmente, se entrenan los algoritmos de aprendizaje reforzado (Aleatorio, DQN y Actor-Critic para los casos con una muestra y múltiples muestras (ver sección 4.6.2)). Para que el agente aprenda una política que considere la incertidumbre de las variables, el entorno utiliza aleatoriamente los escenarios de demanda de energía eléctrica y de caudal de agua (escenarios climáticos) descritos en la sección 3.2. Se selecciona el mejor algoritmo a través de medidas asociadas a RL. Como resultado, se espera obtener una política robusta de operación que considere la incertidumbre climatológica y de demanda. Esta política aprendida se aplica al periodo de estudio 2019-2020, utilizando la demanda y los aportes de agua reales. Como resultado, se obtienen valores de $G_{p,t}$, $Emb_{h,t}$ y $D_{h,t}$ simulados para los 24 meses, los cuales son comparados con información real del sistema colombiano.

En la fase de entrenamiento de los algoritmos de RL, se define como criterio de parada 1500 episodios, entendiendo que cada episodio representa la optimización de 24 meses del horizonte de optimización, para un escenario de aportes y demanda seleccionado aleatoriamente (36000 iteraciones en total). El motivo detrás de este número obedece a una restricción de diseño ocasionado por los tiempos de ejecución del software de optimización (el entorno es lento en su respuesta), que hace que los tiempos de ejecución de los algoritmos sea elevado.

5.2. MEDIDAS DE CALIDAD DEL EXPERIMENTO.

Las medidas de calidad son de dos tipos: Una relacionada con el entrenamiento de los algoritmos de aprendizaje reforzado profundo (DRL), y otra asociada a la comparación de la información entre el caso de estudio 2019-2020 y la obtenida por parte del agente seleccionado en la fase de entrenamiento.

Como medidas para comparar el desempeño de los métodos de DRL en su proceso de entrenamiento, se proponen:

- La Recompensa Promedio Acumulada (*Average Accumulated Reward - AAR*) [44], que permite medir entre algoritmos la mayor recompensa acumulada en el entrenamiento. Se calcula mediante la expresión:

$$AAR = \frac{\sum_{e \in E}(R_1 + R_2 + \cdots + R_e)}{N} \quad (36)$$

  Donde, *E:* conjunto de los cien episodios más recientes, *Re:* es la recompensa obtenida en el episodio e, *N:* número de episodios.

  Para el cálculo del AAR se utiliza una ventana móvil de cien datos, para omitir las recompensas en periodos iniciales donde el agente al estar en etapa de exploración cuenta con bajos niveles de recompensa, generando distorsiones en el cálculo.

- El Mínimo-Máximo Promedio (*MinMax Average - MMAVG*) [44], es una medida que permite valorar la volatilidad del algoritmo respecto a la recompensa promedio, lo que permite tener una noción sobre la estabilidad del algoritmo en el aprendizaje. Para el proyecto se le suma al denominador una constante para evitar problemas de división por cero:

$$MMAVG(i) = \frac{max(i) - min(i)}{AvgReward(i)} \quad (37)$$

  Donde: *i:* Conjunto de datos de los cien episodios más recientes, *AvgReward(i):* Recompensa promedio del conjunto de datos *i, max(i):* máxima recompensa del conjunto de datos i, *min(i):* mínima recompensa del conjunto de datos *i*.

Para comparar la información del caso de estudio 2019-2020 con la simulada, se proponen las siguientes medidas:

- El error porcentual absoluto medio (MAPE): es una métrica que permite medir la distancia entre los valores simulados y los reales.

- El coeficiente de correlación de Pearson (*r*): es una medida de similitud que permite evaluar la relación lineal entre dos variables. Para el proyecto, se espera que los valores simulados coincidan no solo en magnitud, sino también en tendencia. Un valor de 1 representa la similitud ideal.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \widehat{y_t}}{y_t}\right| \qquad r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (38)$$

5.3. ENTRENAMIENTO DEL ALGORITMO DRL

A continuación, se describe el proceso de entrenamiento de cada algoritmo:

5.3.1. Agente Aleatorio:

Es importante recordar que la recompensa está definida como el valor negativo de la función objetivo sumado al nivel de los embalses. Por lo tanto, un valor negativo alejado de cero representa altos costos de operación, por el contrario, valores positivos representan un equilibrio entre bajos costos de operación con altos niveles de embalses. La Figura 13 resume el proceso de entrenamiento del agente aleatorio mediante AAR. Se observa que, seleccionando acciones al azar, el agente solo pudo encontrar como recompensa máxima un valor de 0.0278. El valor promedio de AAR para los últimos cien episodios fue -0.5538.

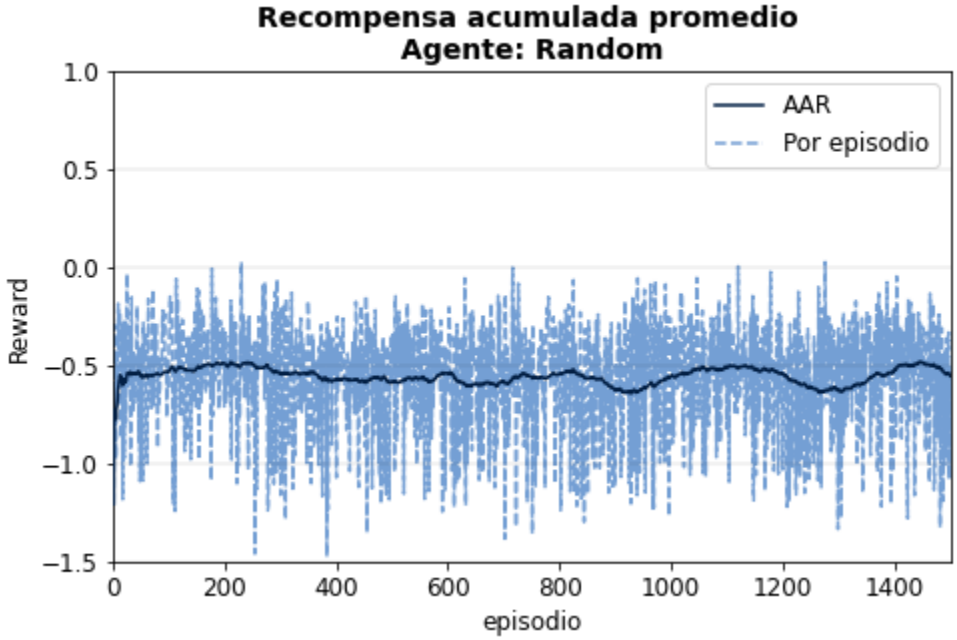


Figura 13. Recompensa puntual y métrica AAR para el proceso de entrenamiento del agento aleatorio. Fuente: Elaboración propia

Por otra parte, la Figura 14 muestra la volatilidad y la carencia de estrategia del agente. Esto se ve reflejado en que no existe ninguna tendencia decreciente a lo largo del entrenamiento, que demuestre el uso sistemático de políticas que aumente el retorno promedio o, por el contrario, la disminución de la volatilidad del rango máximo mínimo. El valor máximo de MMAVG alcanzado fue de 0.43117.

Los resultados anteriores son los que hacen que el agente aleatorio se considere como una buena base de comparación, dado que representa una base o límite inferior para comparar el desempeño de los algoritmos de DRL.

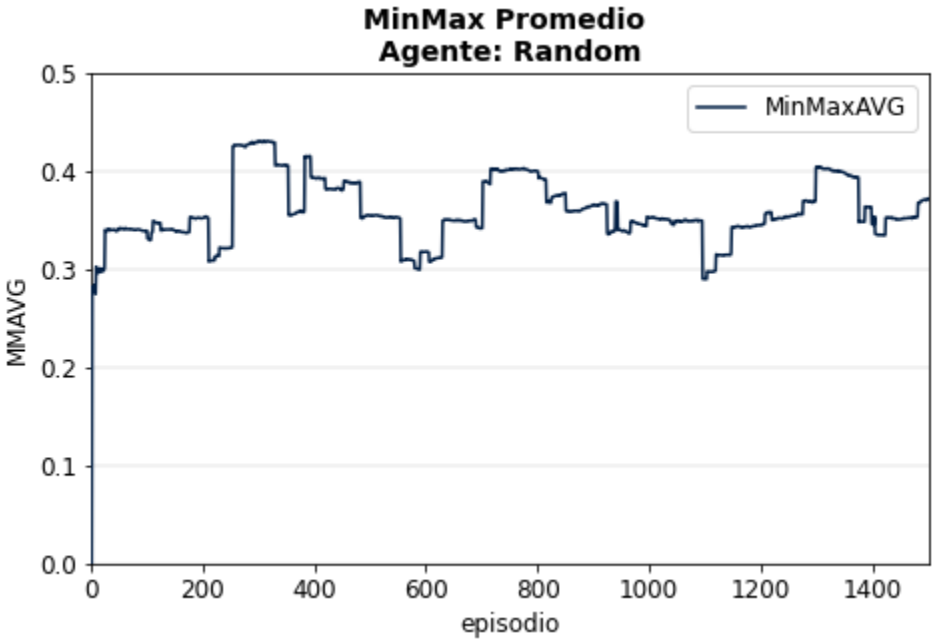


Figura 14. Métrica MinMaxAVG para el proceso de entrenamiento del agento aleatorio. Fuente: Elaboración propia.

5.3.2. Agente DQN:

La Figura 15 resume el proceso de entrenamiento del agente DQN mediante la métrica AAR. Se observa que, comparado con el agente aleatorio, DQN logra encontrar como recompensa máxima un valor de 0.4821. El valor promedio de AAR para los últimos cien episodios fue 0.1111. Visualmente se observa como el agente DQN encuentra de manera sistemática mejores recompensas a medida que avanza el proceso de entrenamiento.

Por otra parte, la Figura 16 refleja el desempeño de acuerdo con la métrica MMAVG. Inicialmente, se observa un comportamiento similar al agente aleatorio hasta el episodio 700, esta semejanza se explica porque en etapas tempranas DQN favorece la exploración. Luego cambia a una tendencia decreciente, lo que muestra que el agente de manera sistemática está aumentando el retorno promedio y disminuyendo la varianza del rango máximo mínimo, asociado a la selección de acciones más ambiciosas (mayor explotación). El valor máximo de MMAVG alcanzado fue de 0.3791, menor que el agente aleatorio, lo que refleja que el entrenamiento de DQN es menos volátil.

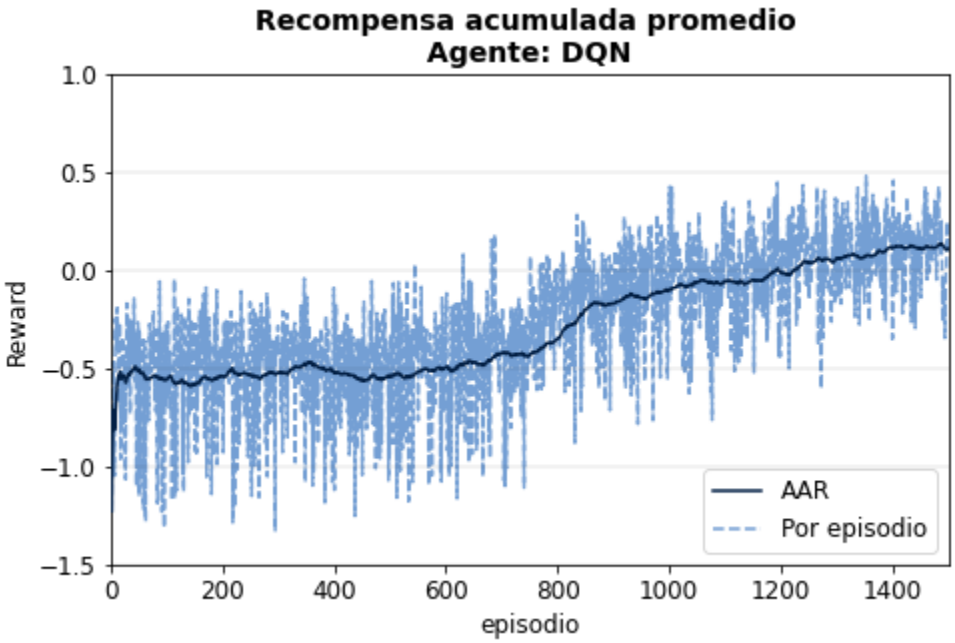


Figura 15. Recompensa puntual y métrica AAR para el proceso de entrenamiento. Fuente: Elaboración propia

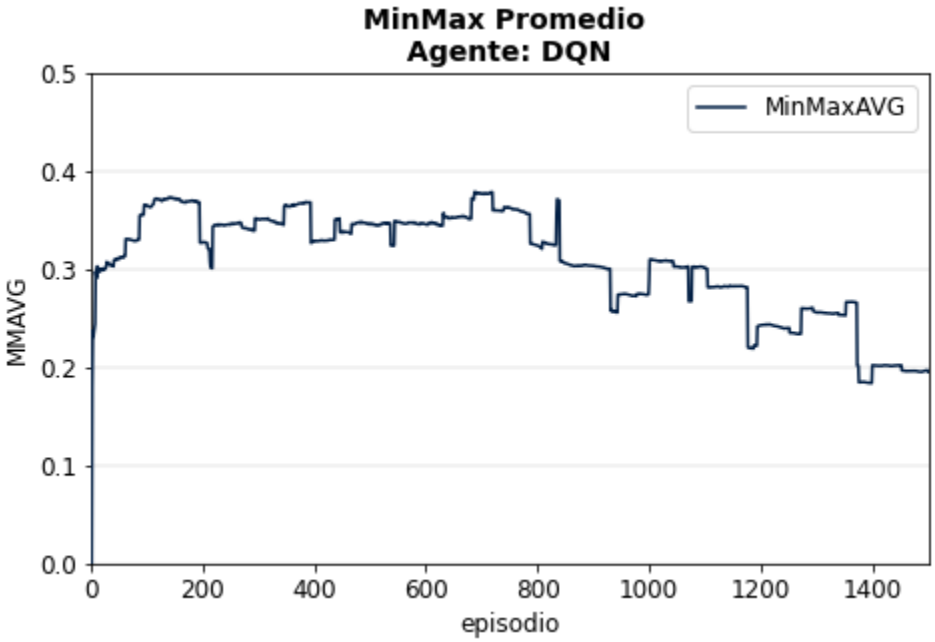


Figura 16. Métrica MinMaxAVG para el proceso de entrenamiento del agento aleatorio. Fuente: Elaboración propia.

5.3.3. Agente A2C:

La Figura 17, resume el proceso de entrenamiento del agente A2C con muestreo simple (A2Cs) mediante la métrica AAR. Se observa que, comparado con el agente aleatorio, A2Cs logra encontrar una mejor recompensa máxima de 0.4914. El valor promedio de AAR para

los últimos cien episodios fue 0.0606. Visualmente se observa como el agente A2Cs encuentra de manera sistemática mejores recompensas a medida que avanza el proceso de entrenamiento.

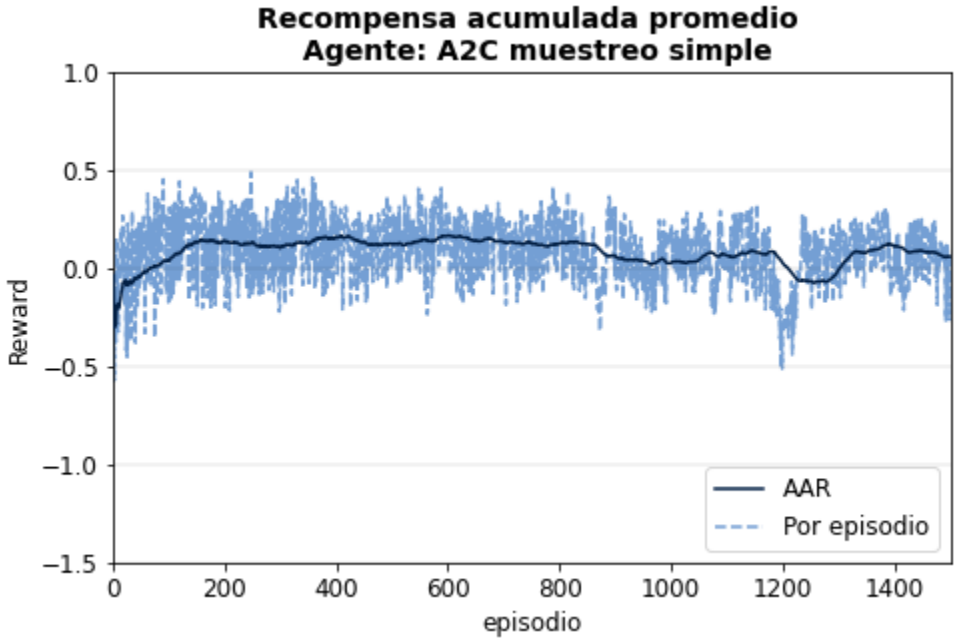


Figura 17. Recompensa puntual y la métrica AAR para el proceso de entrenamiento. Fuente: Elaboración propia

La Figura 18 refleja el desempeño de acuerdo con la métrica MMAVG. Inicialmente se destaca la velocidad del algoritmo, en términos de episodios, en empezar una tendencia decreciente aproximadamente desde el episodio 100, lo que muestra que rápidamente alcanza altos valores de retorno acumulado (como se muestra en la Figura 17) y una menor variación del rango, alcanzado un valor pico de MMAVG de 0.2544, que es mucho menor a lo obtenido por DQN y el aleatorio.

Otro aspecto importante por destacar en las gráficas es la caída del valor AAR aproximadamente en el episodio 1200 (Figura 17), y de forma equivalente, la subida de MMAVG (Figura 18). Esto es un efecto no deseado porque corresponde a un problema de olvido catastrófico por parte de la red neuronal, donde el agente debe aprender nuevamente lo aprendido. Este es un problema característico cuando el entrenamiento de la red neuronal se realiza con un dato al tiempo (una única muestra), lo que dificulta la generalización (en nuestro caso, obtener una política óptima).

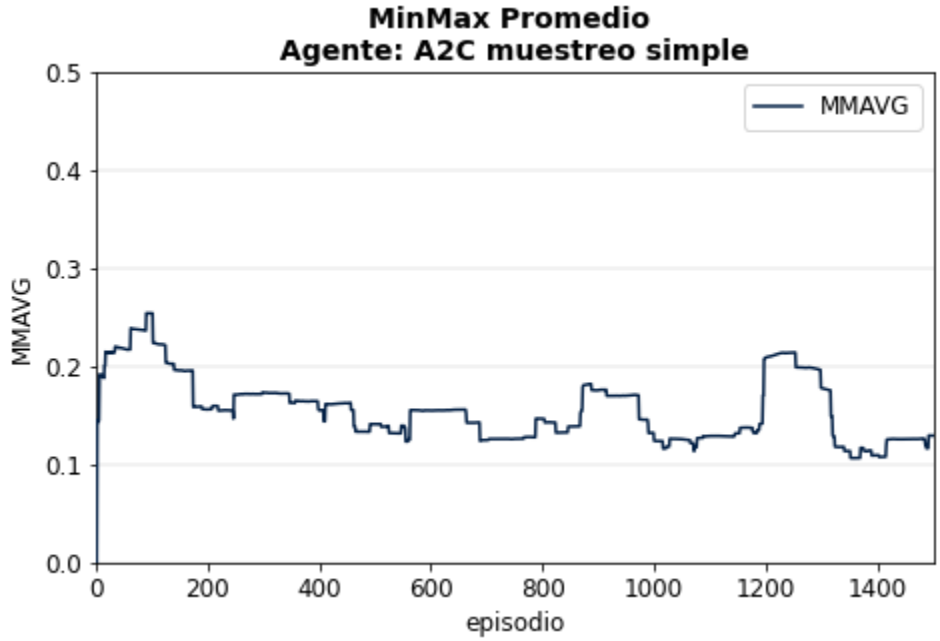


Figura 18. Métrica MinMaxAVG para el proceso de entrenamiento del agento aleatorio. Fuente: Elaboración propia.

Por otra parte, la Figura 19 resume el proceso de entrenamiento del agente A2C con varias muestras (A2Cm) mediante la métrica AAR. Se observa que, comparado con el agente aleatorio, A2Cm logra encontrar una mejor recompensa máxima de 0.5069. El valor promedio de AAR para los últimos cien episodios fue 0.2300. Visualmente se observa como el agente A2Cm encuentra de manera sistemática mejores recompensas a medida que avanza el proceso de entrenamiento.

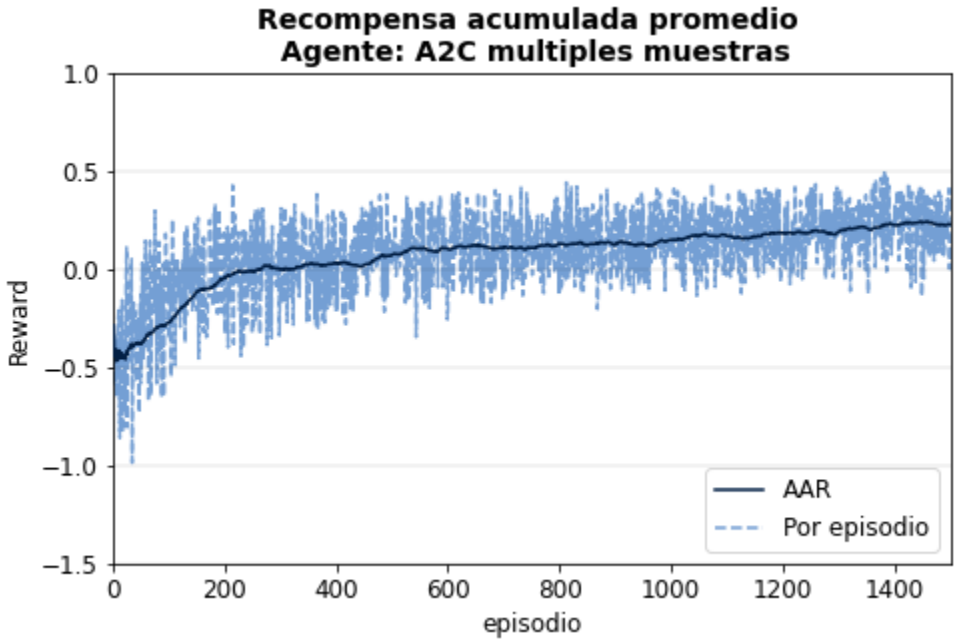


Figura 19. Recompensa puntual y la métrica AAR para el proceso de entrenamiento. Fuente: Elaboración propia

Finalmente, la Figura 20 refleja el desempeño de acuerdo con la métrica MMAVG. Se destaca que al igual que el caso anterior, el agente también encuentra retornos positivos rápidamente (como se muestra en la Figura 19), alcanzando su valor pico de MMAVG de 0.3479 en el episodio 100. Sin embargo, a diferencia de la Figura 18, se observa una tendencia decreciente sostenida a lo largo de los episodios, lo que refleja una mejor estabilidad del algoritmo en sostener mejores retornos promedio y una menor variación en el rango máximo y mínimo.

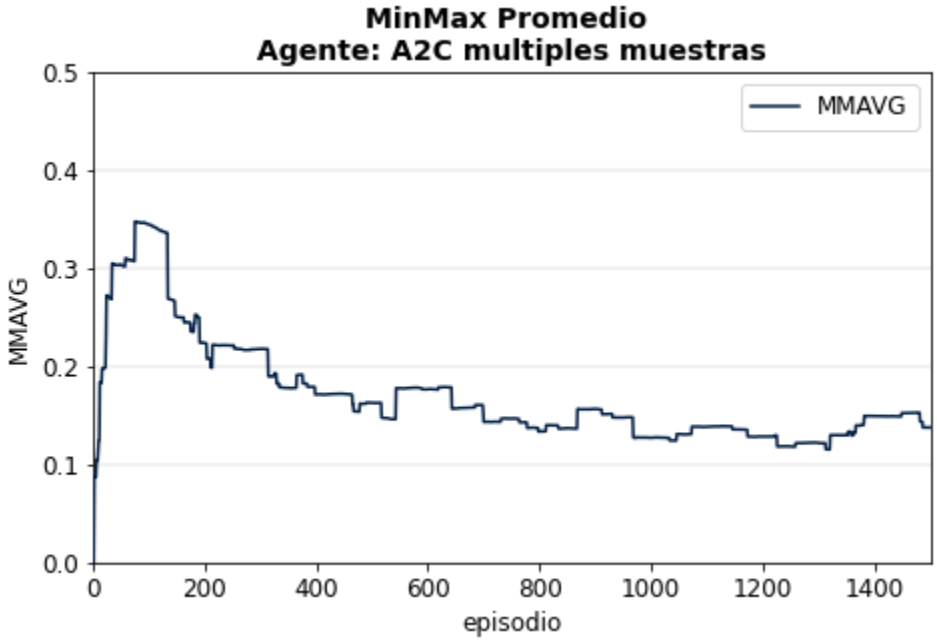


Figura 20. Métrica MinMaxAVG para el proceso de entrenamiento del agento aleatorio. Fuente: Elaboración propia.

5.3.4. Selección del algoritmo.

Para la selección del algoritmo, se comparan los algoritmos mediante las medidas AAR Y MMAVG.

La Figura 21 resume gráficamente la métrica AAR obtenido por cada algoritmo. De acuerdo con la gráfica, los algoritmos A2C encuentran mayores recompensas acumuladas más rápido que DQN en etapas tempranas del entrenamiento. Sin embargo, se observa que A2C con muestreo sencillo se queda atrapado en un óptimo local, haciendo que al final del entrenamiento sea superado por DQN. De acuerdo con la medida AAR, el agente A2C con muestreo múltiple presenta el mejor desempeño.

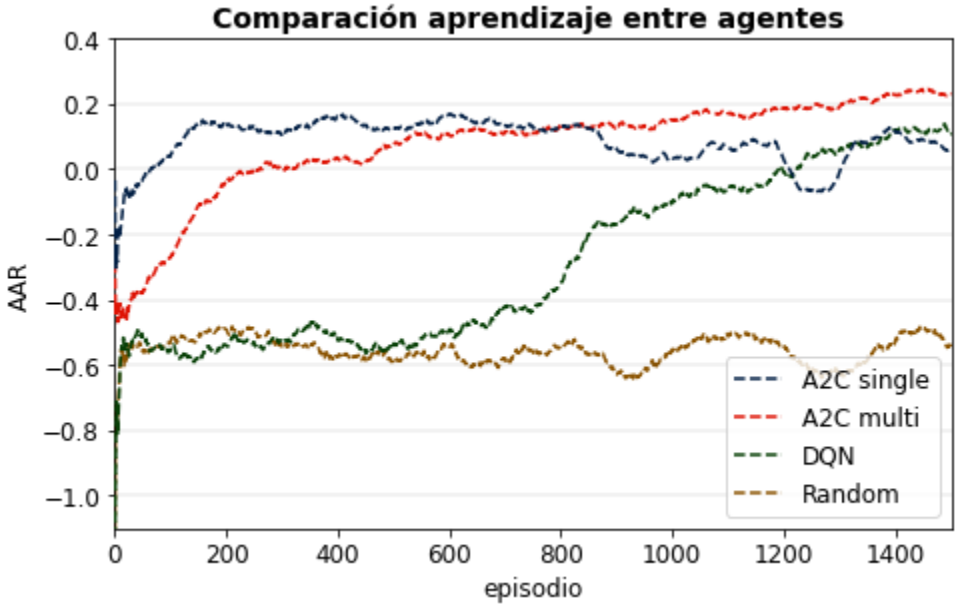


Figura 21. Comparación del aprendizaje entre agentes tomando la métrica AAR. Fuente: Elaboración propia

La Figura 22 muestra que las dos versiones de A2C presentan un desempeño similar al final de los episodios con la métrica MMAVG, sin embargo, hay que destacar que a lo largo del entrenamiento la variante con muestreo es mejor porque no presenta fuertes sobresaltos en el proceso de aprendizaje. Por lo anterior, el mejor algoritmo, considerando tanto AAR como MMAVG, es A2C con muestro.

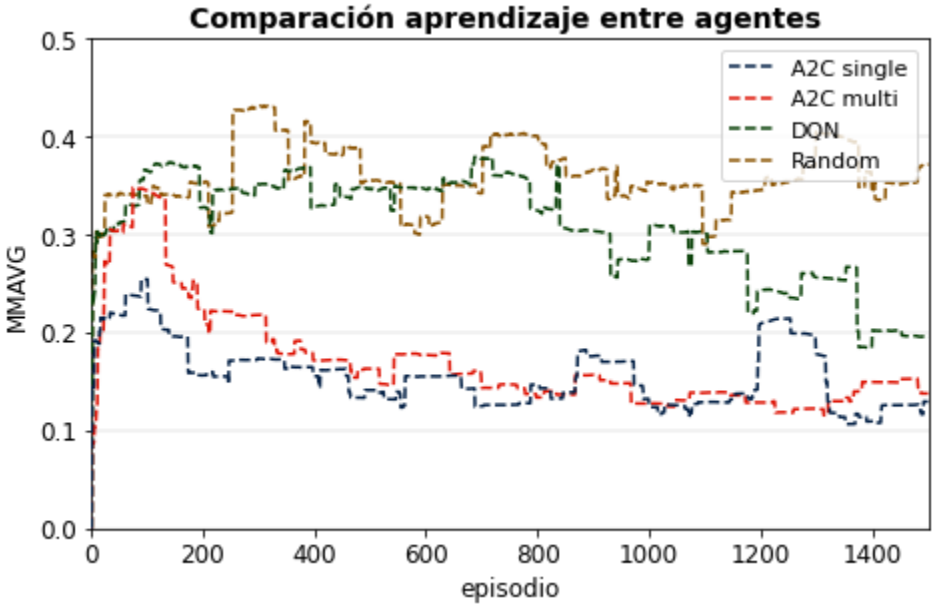


Figura 22. Comparación del aprendizaje entre agentes tomando la métrica MMAVG. Fuente: Elaboración propia

5.4. APLICACIÓN DE LA POLÍTICA AL EL CASO DE ESTUDIO COLOMBIANO

A continuación, se aplica la política aprendida por el agente A2C para el caso de muestreo múltiple al caso de estudio, por ser el algoritmo con mejores resultados. De acuerdo con lo descrito previamente, esta modelación entrega como salida valores de $G_{p,t}$, $Emb_{h,t}$ y $D_{h,t}$ para los 24 meses del periodo de estudio. Luego, estos resultados son comparados con la información real publicada por XM mediante el precio de bolsa y el nivel de los embalses. Adicionalmente, se propone como línea base, simular el caso de estudio usando directamente el modelo matemático descrito en el capítulo 3 (modelo base), para tener una noción del nivel de desempeño que se espera superar.

5.4.1. Precio de bolsa:

El precio de bolsa, como se ha mencionado anteriormente, es un buen indicador para validar las variables $G_{p,t}$ y $D_{h,t}$. La Figura 23 resume los valores obtenidos. Respecto a la información real, se observa que de enero de 2019 a diciembre de 2020 tuvo un valor promedio de 242.23 COP/kWh. Por otra parte, el precio simulado por el modelo base muestra un valor promedio de 341.08 COP/kWh.

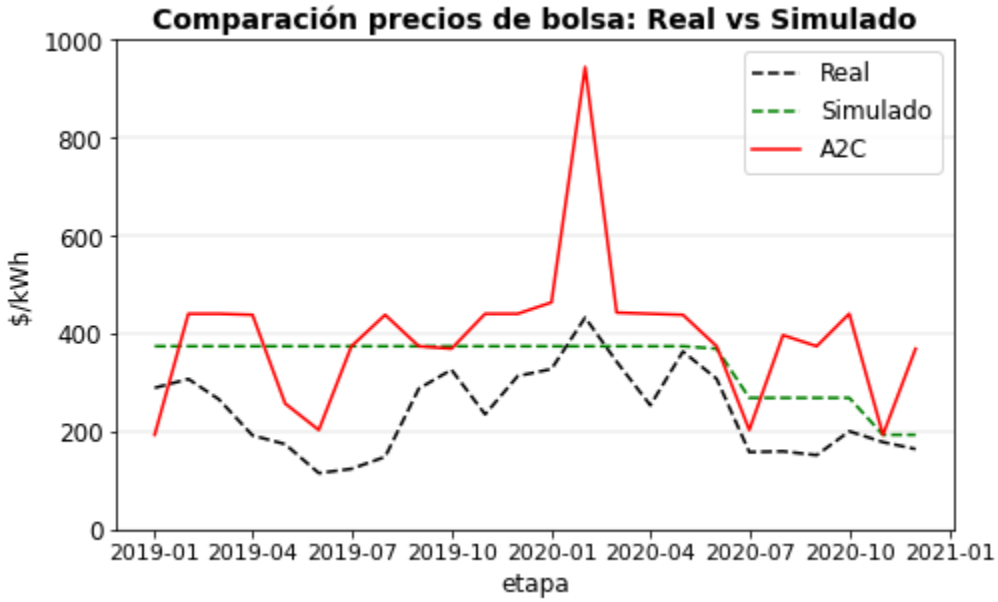


Figura 23. Comparación precios de bolsa. Fuente: Elaboración propia

Respecto al desempeño del agente A2C, se tiene un precio promedio de 395.15 COP/kWh, superior a los anteriores. Las diferencias de precio se deben a la generación de las centrales térmicas. Para el caso puntual de febrero de 2020, la generación térmica real fue de 2123.69 GWh-mes mientras que la del modelo fue de 2722.34 GWh-mes. Esto implica un mayor número de centrales térmicas para abastecer la demanda faltante. La tabla 2 resume las métricas asociadas al precio de bolsa:

Tabla 2. Métricas RMSE y coeficiente de correlación de Pearson comparando información real contra el modelo base y el agente A2C para el precio de bolsa.

Base	Estimado	MAPE	r
Precio de bolsa real	Precio de bolsa modelo base	0.76	0.47
Precio de bolsa real	Precio de bolsa agente RL	0.88	0.59

La métrica MAPE muestra que el modelo base presenta valores más cercanos a los reales, sin embargo, por parte del agente A2C, se destaca como el modelo intenta seguir la dinámica real de los precios, mostrando tendencias de precios en meses similares. En particular, se destaca el precio pico en el mes febrero que, si bien el modelo muestra un valor alto, en realidad fue un mes de alerta por ser particularmente cálido y de bajos aportes de agua (ese pico no lo detecta el modelo base). Esta similitud se confirma con el coeficiente de correlación de Pearson, que es mayor para el agente A2C.

Para disminuir las diferencias en precios se puede modificar el supuesto de precio de oferta de las centrales, cambiando del esquema de precio único promedio por planta a precio promedio de planta por mes, lo que ayuda a representar mejor la dinámica de precios. También es posible disminuir las diferencias a través de la recompensa, donde se puede dar un mayor peso a la función objetivo, o bien definiendo mejores escenarios de caudales.

5.4.2. Gestión de los embalses:

La Figura 24 resume el nivel global de los embalses de Colombia para el periodo del caso de estudio, mientras que la tabla 3 resume las métricas asociadas al nivel de los embalses.

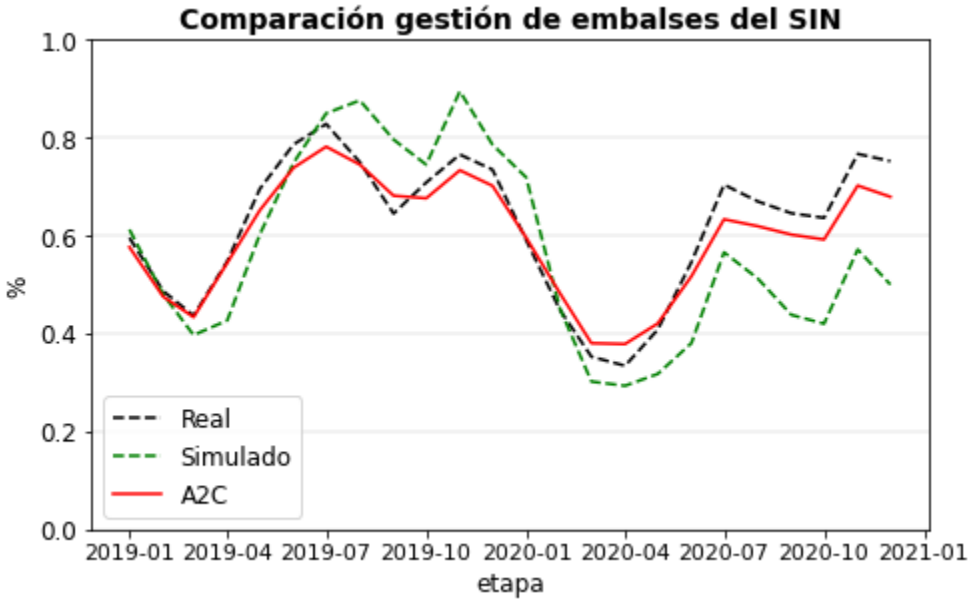


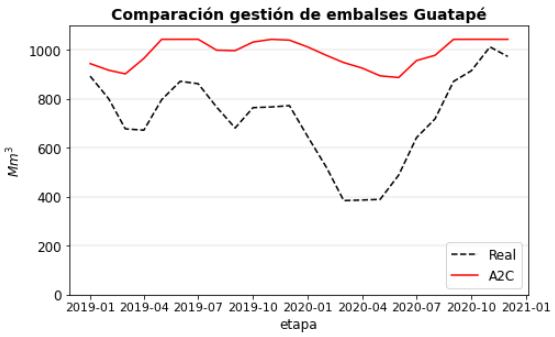
Figura 24. Comparación del nivel de embalse del SIN considerando información real, simulada y la obtenida por el agente RL. Fuente: Elaboración propia

Respecto a la información real, se observa el efecto estacional del clima colombiano sobre el nivel global de los embalses: entre marzo-junio y septiembre-noviembre (temporada de lluvias) se observan valores altos de niveles de embalse, mientras que los meses restantes (temporada seca) se observan bajos niveles. Del modelo base, se destaca como los embalses disminuyen de manera significativa en el 2020, comportamiento que se explica por la función objetivo de la ecuación (17) que busca maximizar el uso del recurso hídrico para minimizar el uso de las centrales térmicas por ser más costosas. Por otro lado, se observa por parte del agente A2C un comportamiento más cercano a la información real (ver tabla 3), lo que refleja la consideración del agente del nivel de los embalses a través de la recompensa.

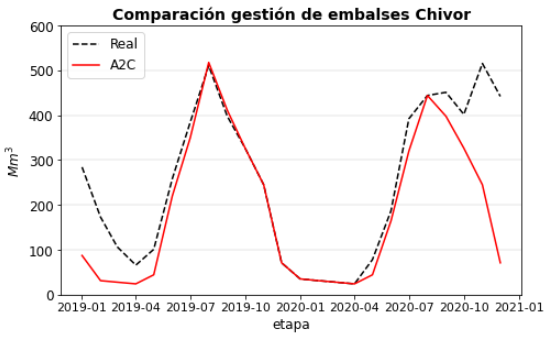
Tabla 3. Métrica MAPE y coeficiente de correlación de Pearson comparando información real contra el modelo base y el agente A2C para el nivel de embalses del sistema.

Base	Estimado	MAPE	r
Nivel de embalse real	Nivel de embalse simulado con información perfecta	0.41	0.78
Nivel de embalse real	Nivel de embalse simulado por agente RL	0.23	0.98

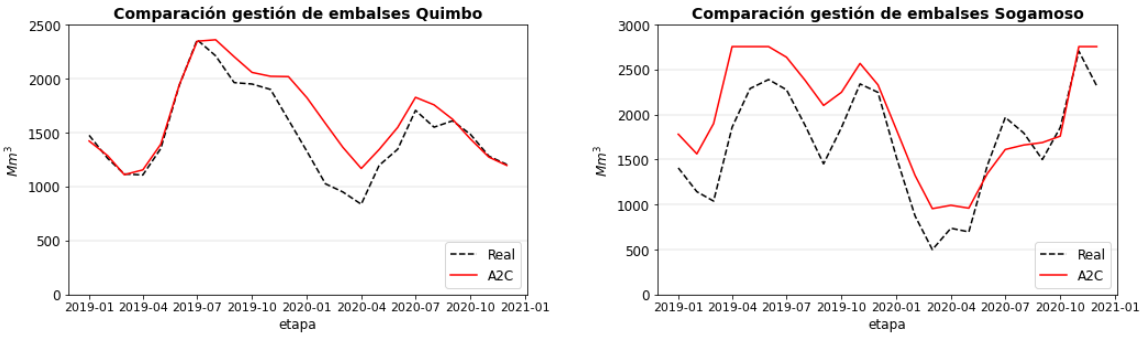
Tanto los valores de MAPE y *r* son mejores para el agente A2C, lo que refleja una mejor captura de la dinámica de los embalses del sistema colombiano. Otra forma de visualizar el desempeño del agente A2C es a través de la Figura 25, donde se observa en detalle la gestión de cuatro importantes embalses del país.



(a)



(b)



(c) (d)

Figura 25. Comparación de embalses en millones de metros cúbicos (Mm3) para (a) Guatapé, (b) Chivor, (c) Quimbo y (d) Sogamoso. Fuente: Elaboración propia

De la Figura 25, se puede observar cómo A2C para los embalses Quimbo (c) y Sogamoso (d) sigue en buena medida la dinámica real del sistema, donde se refleja un seguimiento de las subidas y caídas en los niveles de los embalses a lo largo de los meses. Respecto a Chivor (b), se destaca un buen nivel de ajuste hasta mediados del 2020, donde luego se observa una influencia marcada principalmente por la función objetivo, tal que el agente busca usar el mayor recurso hídrico posible de algunos embalses. Finalmente, respecto a Guatapé (a), se observa que el agente busca mantener dicho embalse en niveles altos, lo que es interesante porque al parecer el agente usa esta central como holgura de agua ante posibles escenarios secos, como sucedió en el primer trimestre del 2020.

CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS

El problema de despacho energético económico hidro-térmico es un problema que puede ser modelado bajo un esquema de MDPs, lo que permite utilizar algoritmos de aprendizaje reforzado para su solución, enriqueciendo el análisis tradicional al permitir incluir la incertidumbre de variables a través del entorno. Para el caso de la presente propuesta, fue necesario el diseño completo del entorno, considerar el manejo de acciones simultáneas por parte del agente en el entorno, y el uso de los algoritmos DQN (limitado para acciones discretas) y A2C. Lo anterior, exige una alta inversión de tiempo en experimentación para calibrar los parámetros que logren un desempeño acorde a lo esperado.

De acuerdo con las medidas establecidas en RL, el proceso de experimentación en el caso de estudio colombiano muestra que el algoritmo A2C tiene mejor desempeño que DQN en recompensa acumulada y velocidad de aprendizaje. Aunque DQN muestra una tendencia de que puede seguir encontrando mejores recompensas, sería necesario incrementar el número de episodios, lo que supone una gran desventaja comparado con los algoritmos A2C que logran encontrar mejores valores en un menor número de episodios. Por parte de la métrica MMAVG, se observa que el algoritmo A2C con muestreo sencillo no es estable a lo largo del entrenamiento. El mejor algoritmo respecto a AAR y MMAVG es A2C con muestreo múltiple, el cual tuvo la mejor recompensa acumulada promedio y menor variabilidad en pocos episodios.

Respecto a la aplicación de la política aprendida, para el caso del precio de bolsa se observa que el agente logra una operación cuya dinámica sigue la tendencia de precios real, pero alejado en magnitud. Lo anterior es posible corregirlo mejorando los supuestos iniciales en los precios de oferta, o bien mejorando la definición de los escenarios de caudales. Respecto a la gestión de los embalses, si se observa un mejor desempeño tanto en tendencia como en magnitud. Lo anterior permite concluir como a través del aprendizaje reforzado es posible incluir en el análisis del despacho económico la incertidumbre de variables tan importantes como la demanda y los aportes de agua.

Finalmente, para los trabajos futuros, una propuesta consiste en incluir en el análisis la expansión del sistema de generación, donde el reto es modelar el dinamismo en el número de centrales en el tiempo, lo que supone inicialmente cambios en el diseño del agente, el entorno, el estado y la recompensa, porque fueron diseñados bajo el supuesto de un número fijo de centrales. Otra propuesta consiste en cambiar la arquitectura del agente, tal que las acciones pasarían de ser centralizadas por un agente a un enfoque de varios agentes en competencia, lo que resulta apropiado teniendo en cuenta que las centrales

hidroeléctricas de Colombia están bajo un marco de mercados en competencia. Para esto, se tendría que incluir en la modelación conceptos de la teoría de juegos y sistemas multiagentes que permitan a un agente tomar acciones teniendo en cuenta las decisiones de los demás. La idea sería validar si este enfoque mejora el desempeño, teniendo en cuenta que se modelan más aspectos del mercado colombiano como, por ejemplo: el interés económico de los agentes generadores. A nivel de técnicas de aprendizaje profundo reforzado, posibles trabajos serian aplicar otros algoritmos como: Asynchronous Actor-Critic (A3C), Soft Actor-Critic (SAC) y Deep Deterministic Policy Gradient (DDPG), los cuáles consideran otros aspectos en el proceso de aprendizaje y, por tanto, pueden eventualmente dar mejores resultados.

[1] N. Yan, Z. X. Xing y B. Zhang, «Economic Dispatch Application of Power System with Energy Storage Systems,» *IEEE Transactions on Applied Superconductivity,* vol. 26, nº 7, pp. 1-5, 2016, doi: 10.1109/TASC.2016.2598963.

[2] PSR - Energy Consulting and Analytics, «SDDP Manual de Metodología,» Marzo 2018. [En línea]. Available: https://www.asep.gob.pa/wp-content/uploads/electricidad/consultas_publicas/2019/cp_012-2019/tomoII-plan_indicativo_generacion_2019-2033/tomoII-anexo_9-metodologia_modelos_Optgen_SDDP.pdf. [Último acceso: Octubre 2021].

[3] H. Yuping, P. Panos y Z. Qipeng, Electrical Power Unit Commitment, Boston, MA: Springer Briefs in Energy., 2017.

[4] Compañía Expertos en Mercados XM, «Informe General del Mercado Mayo 2021,» Junio 2021. [En línea]. Available: https://www.xm.com.co/nuestra-empresa/informes/informes-de-la-operacion-y-el-mercado/informes-mensuales-de-analisis-del-mercado. [Último acceso: Agosto 2021].

[5] M. Pereira y L. Pinto, «Stochastic Optimization of a Multireservoir Hydroelectric System: A Descomposition Approach,» *Water Resources Research,* vol. 21, nº 6, pp. 779-792, 1985.

[6] J. K. Pattanaik, M. Basu y D. P. Dash, «Review on Application and Comparison of Metaheuristic Techniques to Multi-area Economic Dispatch Problem,» *Protection and Control of Moder Power Systems,* vol. 2, nº 17, pp. 1-11, 2017, doi: 10.1186/s41601-017-0049-x.

[7] M. Löschenbrand y M. Korpas, «Multiple Nash Equilibria in Electricity Markets with Price-Making Hydrothermal Producers,» *IEEE Transactions on Power Systems,* vol. 34, nº 1, pp. 422-431, 2019, doi: 10.1109/TPWRS.2018.2858574.

[8] R. Sutton y A. Barto, Reinforcement Learning. An Introduction, Cambridge, Massachusetts: The MIT Press, 2018.

[9] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam y et. al, «Mastering the Game of Go with Deep Neural Networks and Tree Search,» *Nature,* vol. 529, pp. 484-503, 2016, doi: 10.1038/nature16961.

[10] V. Mnih, K. Kavukcouglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra y M. Riedmiller, «Playing Atari with Deep Reinforcement Learning,» *NIPS Deep Learning Workshop,* pp. 1-9, 2013, doi: 10.48550/arXiv.1312.5602.

[11] W. Powell, From Reinforcement Learning to Optimal Control: A unified framework for sequential decisions, Pricenton, NJ: Department of Operation Research and Financial Engineering., 2019.

[12] M. A. Rastegar, E. Guerci y S. Cincotti, «Agent-Based Modelo of the Italian Wholesale Electricity Market,» *International Conference on the European Electricity Market,* pp. 1-7, 2009, doi: 10.1109/EEM.2009.5207128.

[13] R. Ragupathi y T. Das, «A Stochastic Game Approach for Modeling Wholesale Energy Bidding in Deregulated Power Markets,» *IEEE Transactions on Power Systems.,* vol. 19, nº 2, pp. 849-856, 2004, doi: 10.1109/TPWRS.2004.825910.

[14] A. Mallem y O. Boudebbouz, «Economic Dispatch on a Power System Network Interconnected with Solar Farm,» *International Conference on Sustainable Renewable Energy System and Applications (ICSRESA),* pp. 1-6, 2019, doi: 10.1109/ICSRESA49121.2019.9182477.

[15] Z. Liu, B. Tessema, G. Papaefthymiou y L. v. d. Sluis, «Transmission Expansion Planning for Congestion Allevation Using Constrained Location Marginal Price,» *IET Conference on Reliability of Transmission and Distribution Networks (RTDN),* pp. 1-6, 2011, doi: 10.1049/cp.2011.0538.

[16] S. Ohnishi, E. Uchibe, Y. Yamaguchi, Y. Yasui y S. Ishii, «Constrained Deep Q-Learning Gradually Approaching Ordinary Q-Learning,» Frontiers in Neurobotics, China, 2019.

[17] Optimization, «Oxford Reference,» [En línea]. Available: https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100252326. [Último acceso: 28 09 2021].

[18] I. Grondman, L. Busoniu, G. Lopes y R. Babuska, «A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients,» *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews),* vol. 42, nº 6, pp. 1291-1307, 2012, doi: 10.1109/TSMCC.2012.2218595.

[19] L. Yu y N. Li, «A Reinforcement Learning Algorithm based on Neural Network for Economic Dispatch,» *Chinese Control Conference (CCC),* pp. 1637-1642, 2020, doi: 10.23919/CCC50068.2020.9188641.

[20] B. Sallans y G. Hinton, «Reinforcement Learning with Factored State and Actions,» *Journal of Machine Learning Research,* vol. 5, pp. 1063-1088, 2004.

[21] H. Zhu y M. Kirley, «Deep Multi-agent Reinforcement Learning in a Common-Pool Resource System,» *IEEE Congress on Evolutionary Computation (CEC),* pp. 142-149, 2019, doi: 10.1109/CEC.2019.8790001.

[22] Y. Chen, F. Zhang y Z. Liu, «Adaptive Advantage Estimation for Actor-Critic Algorithms,» *International Joint Conference on Neural Networks (IJCNN),* pp. 1-8, 2021, doi: 10.1109/IJCNN52387.2021.9534005..

[23] C.-G. Li, M. Wang y Q.-N. Yuan, «A Multi-Agent Reinforcement Learning Using Actor-Critic Methods,» *International Conference on Machine Learning and Cybernetics,* pp. 878-882, 2008, doi: 10.1109/ICMLC.2008.4620528.

[24] Compañía Expertos en Mercados XM S.A E.S.P, «Economic Dispatch Model of the Colombian Electricity System,» ILOG'S Premier User Conference (DIALOG09), Orlando, 2009.

[25] A. Ramos, «Optimización Estocástica,» Universidad Pontificia Comillas, 2016. [En línea]. Available: https://pascua.iit.comillas.edu/aramos/simio/apuntes/a_sp.pdf. [Último acceso: Abril 2021].

[26] A. Shapiro, «Analysis of Stochastic Dual Dynamic Programming Method,» *European Journal of Operational Research,* vol. 209, nº 1, pp. 63-72, 2011, doi: 10.1016/j.ejor.2010.08.007.

[27] V. Charles, S. Ansari y M. Khalid, «Multi-Objective Stochastic Linear Programming with General form of Distributions,» *International Journal of Operations Research and Optimization,* vol. 2, pp. 261-278, 2011.

[28] H. Abdi, H. Fattahi y S. Lumbreras, «What Metaheuristic Solves the Economic Dispatch Faster?: a comparative case study,» *Springer. Electrical Engineering,* nº 100, pp. 2825-2837, 2018, doi: 10.1007/s00202-018-0750-4.

[29] H. Kishan, K. Jain y M. Pandit, «Performance Analysis of Metaheuristic Technique for Noncovex Economic Dispatch,» *International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2011),* pp. 396-402, 2011, doi: 10.1049/cp.2011.0396.

[30] T. Krause, E. Beck, R. Cherkaoui, A. Germond, G. Andersson y D. Ernst, «A comparison of Nash Equilibria Analysis and Agent-Based Modelling for Power Markets,» *Electrical Power and Energy Systems,* vol. 28, pp. 599-607, 2006, doi:10.1016/j.ijepes.2006.03.002.

[31] M. Carvalho y J. Pedroso, «Electricity Day-Ahead Markets: Computation of Nash Equilibria,» *Journal of Industrial and Management Optimization,* vol. 11, nº 3, pp. 985-998, 2015, doi: 10.3934/jimo.2015.11.985.

[32] S. de la Torre, J. Contreras y A. Conejo, «Finding Multiperiod Nash Equilibria in Pool-Based Electricity Markets,» *IEEE Transactions on Power Systems,* vol. 19, nº 1, pp. 643-651, 2004, doi: 10.1109/TPWRS.2003.820703.

[33] E. A. Jasmin, T. P. Imthias Ahamed y V. Jagathiraj, «A Reinforcement Learning algorithm to economic dispatch considering transmission losses,» *TENCON 2008. IEEE Region 10 Conference,* pp. 1-6, 2008, doi: 10.1109/TENCON.2008.4766652.

[34] I. A. T. Parambath, E. A. Jasmin y E. A. Ai-Ammar, «Reinforcement learning solution to economic dispatch using pursuit algorithm,» *IEEE GCC Conference and Exhibition (GCC),* pp. 263-266, 2011, doi: 10.1109/IEEEGCC.2011.5752517.

[35] T. P. Imthias Ahmed y E. A. Jasmin, «Reinforcement Learning solution for economic scheduling with stochastic cost function,» *2011 IEEE Recent Advances in Intelligent Computational Systems,* pp. 437-440, 2011, doi: 10.1109/RAICS.2011.6069350.

[36] M. Abouheaf, S. Haesaert, W.-J. Lee y L. Lewis, «Approximate and Reinforcement Learning techniques to solve non-convex Economic Dispatch problems,» *IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14),* pp. 1-8, 2014, doi: 10.1109/SSD.2014.6808789.

[37] M. Ali, E. Guerci y S. Cincotti, «Agent-based model of the Italian wholesale electricity market,» *International Conference on the European Energy Market,* pp. 1-7, 2009, doi: 10.1109/EEM.2009.5207128.

[38] L. Ding, Z. Lin, X. Shi y G. Yan, «Target-Value-Competition-Based Multi-agent Deep Reinforcement Learning Algorthm for Distributed Nonconvex Economic Dispatch,» *IEEE Transactions on Power Systems,* pp. 1-14, 2022.

[39] Y. Shu, W. Bi, W. Dong y Q. Yang, «Dueling Double Q-Learning based Real-time Energy Dispatch in Grid-connected Microgrids,» *International on Distgributed Computing and Applications for Business Engineering and Science,* pp. 42-45, 2020.

[40] R. S, J. M. Johnson y T. P. Imthias, «Short Term Hydrothermal Scheduling Using Reinforcement Learning,» *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS),* pp. 1-6, 2019.

[41] L. Gallego, O. Duarte y A. Delgadillo, «Strategic bidding in Colombian electricity market using a multi-agent learning approach,» *IEEE/PES Transmission and Distribution Conference and Exposition: Latin America,* pp. 1-7, 2008, doi: 10.1109/TDC-LA.2008.4641706.

[42] J. Kok y N. Vlassis, «Collaborative Multiagent Reinforcement Learning by Payoff Propagation,» *Journal of Machine Learning Research,* vol. 7, pp. 1789-1828, 2006.

[43] N. Balachandar, J. Dieter y G. Sachithanandam, «Collaboration of AI Agents via Cooperative Multi-Agente Deep Reinforcement Learing,» *Computing Research Repository (CoRR),* pp. 1-9, 2019, doi: 10.48550/arXiv.1907.00327.

[44] A. Oroojlooy y D. Hajinezhad, «A Review of Cooperative Multi-Agent Deep Reinforcement Learning,» *Computer Science,* pp. 1-81, 2019, doi: 10.48550/arXiv.1908.03963.

[45] T. Théate y D. Ernst, «An Application of Deep Reinforcement Learning to Algorithmic Trading,» *Expert Systems with Applications,* vol. 173, p. 19, 2021, doi: 10.48550/arXiv.2004.06627.

[46] W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland y W. Dabney, «Revisting Fundamentals of Experience Replay,» *Proceedings International Conference on Machine Learning,* pp. 1-19, 2020, doi: 10.48550/arXiv.2007.06700.

[47] B. Oliveira, C. Martins y F. Magalhaes, «Difference Based Metrics for Deep Reinforcement Learning,» *IEEE Access,* vol. 7, pp. 159141-159149, 2019, doi: 10.1109/ACCESS.2019.2945879.

[48] S. Marín, «Ética e Inteligencia Artificial,» Cátedra CaixaBank de Responsabilidad Social Corporativa, Navarra, 2019, doi: 10.15581/018.ST-522.

DESCRIPCIÓN Y ADQUISICIÓN DE LOS DATOS.

Para el desarrollo de la propuesta en la construcción del caso de estudio y la creación de los escenarios de demanda y caudal requeridos para diseño del entorno, utiliza como principal fuente de información es el Portal SINERGOX[11] de XM, donde se consulta la información asociada a la hidrología, demanda y centrales eléctricas. La cadena hidrológica de los embalses está disponible en varios estudios de la Unidad de Planeación Minero-Energética (UPME[12]), que son de libre acceso. La información para calcular los factores de conversión para convertir caudal en energía se puede consultar en los acuerdos del Consejo Nacional de Operación (CNO[13]). Toda la información descrita es de libre acceso. En la tabla 4 se resumen las fuentes de los datos utilizados:

Tabla 4. Descripción de los datos empleados para el modelo matemático.

Descripción	Adquisición y condiciones de uso	Descripción del archivo	Propósito
Aportes mensuales de caudales de agua.	Descarga desde página web de XM[14]. (Libre acceso)	Archivos .xlsx, con datos mensuales (819 KB)	Hace parte de los parámetros del despacho económico.
Capacidad efectiva neta.	Descarga desde página web de XM[15]. (Libre acceso)	Archivos .xlsx, con datos diarios (4012 KB)	Hace parte de los parámetros del despacho económico.

[11] El portal Sinergox es un sitio web administrado por XM, que centraliza la información pública del Mercado de Energía Mayorista (MEM) y la operación del Sistema Interconectado Nacional (SIN), apoya la toma de decisiones de los diferentes públicos de interés y promueve la competencia y transparencia en el mercado. Disponible en: http://sinergox.xm.com.co/Paginas/Home.aspx

[12] La Unidad de Planeación Minero Energética -UPME es una Unidad Administrativa Especial del orden Nacional, de carácter técnico, adscrita al Ministerio de Minas y Energía -MME, que tiene por objeto planear en forma integral, indicativa, permanente y coordinada con los agentes del sector minero energético, el desarrollo y aprovechamiento de los recursos mineros y energéticos; producir y divulgar la información requerida para la formulación de política y toma de decisiones; y apoyar al MME en el logro de sus objetivos y metas

[13] El Consejo Nacional de Operación -CNO del sector eléctrico, creado por la Ley 143 de 1994 en su artículo 36, es un organismo privado que tiene como función principal acordar los aspectos técnicos para garantizar que la operación del Sistema Interconectado Nacional sea segura, confiable y económica y ser el ejecutor del Reglamento de Operación.

[14] Disponible en: http://sinergox.xm.com.co/hdrlg/Paginas/Historicos/Historicos.aspx

[15] Disponible en: http://sinergox.xm.com.co/oferta/Paginas/Historicos/Historicos.aspx

Descripción	Adquisición y condiciones de uso	Descripción del archivo	Propósito
Demanda de energía.	Descarga desde página web de XM[16]. (Libre acceso)	Archivos .xlsx, con datos diarios (230 KB)	Hace parte de los parámetros del despacho económico.
Reserva mensual de los embalses.	Descarga desde página web de XM[17]. (Libre acceso)	Archivos .xlsx, con datos mensuales (465 KB)	Hace parte de los parámetros del despacho económico. También son los datos para la validación del algoritmo de RL.
Generación de las centrales.	Descarga desde página web de XM[18]. (Libre acceso)	Archivos .xlsx, con datos horarios (28500 KB)	Hace parte de los parámetros del despacho económico. También son los datos para la validación del algoritmo de RL.
Precio de bolsa.	Descarga desde página web de XM[19]. (Libre acceso)	Archivos .xlsx, con datos diarios (44.2 KB)	Hace parte de los datos para la validación del algoritmo de RL.
Factor de descarga de los embalses	Descarga desde la página web del CNO[20]. (Libre acceso)	Archivos .pdf, se toma un parámetro único para todo el horizonte (1750 KB)	Hace parte de los parámetros del despacho económico.
Cadenas hidrológicas	Descarga desde la página de la UPME. (Libre acceso)	Archivos .pdf, se toma un parámetro único para todo el horizonte (4380 KB)	Hace parte de los parámetros del despacho económico.

Se observa que la información está disponible en diferentes escalas de tiempo, sin embargo, toda la información se convierte a su equivalente mensual que es la resolución de tiempo de interés para la investigación. Es importante aclarar que, a pesar de la disponibilidad de la información, no es la mejor disponible y se pueden tener deficiencias en algunos parámetros en la modelación como lo son las cadenas hidrológicas entre embalses, de las cuáles no se tiene detalle en la información, y se modela mediante aproximaciones.

[16] Disponible en: http://sinergox.xm.com.co/dmnd/Paginas/Historicos/Historicos.aspx
[17] Disponible en: http://sinergox.xm.com.co/hdrlg/Paginas/Historicos/Historicos.aspx
[18] Disponible en: http://sinergox.xm.com.co/oferta/Paginas/Historicos/Historicos.aspx
[19] Disponible en: http://sinergox.xm.com.co/trpr/Paginas/Historicos/Historicos.aspx
[20] Disponible en: https://www.cno.org.co/acuerdos

En cuanto a datos faltantes, existe una dificultad con la construcción de escenarios de caudales usando información histórica, porque las centrales nuevas solo tienen información a partir de su entrada en operación al sistema. Sin embargo, dado que son pocas centrales con esta situación, la reconstrucción de las series se realiza mediante análogos climáticos teniendo en cuenta la anomalía de temperatura superficial del mar.

ASPECTOS ÉTICOS.

El potencial que exhibe la inteligencia artificial y la capacidad que tienen los modelos de aprendizaje automático para mejorar su desempeño de una tarea determinada con base en la experiencia [45], hace necesario que se deban tener presentes los riesgos que trae consigo el beneficio obtenido por el desarrollo del presente proyecto:

- **Destrucción de puestos de trabajo:** El presente proyecto se concibe como una herramienta más entre las disponibles para apoyar la toma de decisiones del analista experto. Comparado con las herramientas disponibles existentes tiene la novedad de poder incluir en los análisis la incertidumbre y decisiones operativas, que generalmente se analizan de manera independiente. Por lo anterior, no se genera destrucción de puestos de trabajo dado que aún se requiere del conocimiento experto para entregar las entradas del modelo y la interpretación de los resultados.

- **Respeto a la autonomía humana:** El proyecto respeta la autonomía humana. El modelo de RL carece de autonomía propia que pueda afectar los derechos humanos fundamentales o generar algún tipo de discriminación.

- **Transparencia**: La transparencia está asociada principalmente a la explicabilidad y trazabilidad del sistema. Cómo se ha mencionado, los algoritmos de RL son un marco general para resolver problemas fundamentados en los procesos de decisión de Markov. Aunque de manera general es posible comprender el funcionamiento del algoritmo, una trazabilidad y comprensión absoluta no es posible debido a las abstracciones del modelo, manejo de incertidumbre y de redes neuronales.

- **Responsabilidad y rendición de cuentas**: Debido al alcance investigativo del proyecto que está enfocado en la solución de un modelo matemático con información histórica, no es necesario establecer un esquema o mecanismos de rendición de cuentas ante posibles daños y perjuicios derivados por la ejecución del algoritmo de RL.

- **Robustez y seguridad:** No se tiene contemplado en el diseño esquemas de protección ante ciberataques y fallos técnicos debido al alcance investigativo del proyecto.

- **Justicia y no discriminación:** El proyecto se alimenta de información histórica del sector eléctrico, por lo tanto, no hay uso injusto de los datos que puedan llevar a discriminaciones o distorsiones en los precios o en el equilibrio del mercado.

- **Lucro:** Dado que el proyecto es un ejercicio investigativo en la aplicación de algoritmos de RL en el sector eléctrico, es posible que tanto las empresas prestadoras de servicio, entes gubernamentales, o empresas de consultoría se vean beneficiadas al contar con nuevos modelos para sus análisis o venta.

ALGORITMO ALEATORIO.

Tabla 5. Algoritmo Random para resolver el despacho energético económico

Algoritmo 2 *Algoritmo Random*

Entrada: *estado, acción.*
Parámetros: *parámetro de exploración $\in$, parámetro de aprendizaje,*
Salida: *acción óptima para el tiempo t.*

```
Get Economic Dispatch parameters
Initialize memory 𝒟 to capacity T
Initalize Economic Dispathc model

set ∈=1 # always explore

for each episode I do:
    Initialize state s_0
    for t=1....T_I in episode I do:
        action = call ∈-greedy Algorithm
        execute action a_t and observe reward r_t and state s_{t+1}
        if r_t > historic reward r'_t:
            Store transition
```

ALGORITMO DQN.

Tabla 6. Hiperparámetros DQN

Parámetro	***Valor***
1: Optimizador	*ADAM*
2: Máximo número de pasos	*37*
3: Tasa de aprendizaje	*N.A*
4: Descuento(γ)	*0.9*
5. Tamaño Replay Buffer (𝒟)	*2400*
6: Número de capas ocultas	*2*
7: Número de unidades por capa	*256*
8: Número salidas	*28 centrales por 11 acciones*
9: Número de muestras por lote	*32*
10: Función de activación entrada	*RELU*
11: Función de activación capas ocultas	*ELU*
12: Tasa aprendizaje red neuronal	*0.00025*
13: epsilon-greedy	*10% exploración*
14:Pasos de exploración	*9/10 del total de episodios*

Tabla 7. Algoritmo DQN para resolver el despacho energético económico

Algoritmo 3 *Algoritmo DQN*

Entrada: *estado, acción.*
Parámetros: *parámetro de exploración ∈, parámetro de aprendizaje.*
Salida: *política óptima.*

Get Economic Dispatch parameters
Initialize replay memory $\mathcal{D}$ to capacity N
Initalize Economic Dispatch model
Initialize Q network (θ)
Initialize target Q network (θ') with θ weights

for *each episode I* ***do****:*
  Initialize state s_0
  for *t=1….T_I in episode I* ***do:***
    action = ***call*** *∈-greedy Algorithm*
    execute action a_t and observe reward r_t and state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  if *size batch < $\mathcal{D}$ size:*
    Sample random batch (s_j, a_j, r_j, s_{j+1}) uniformly from $\mathcal{D}$
    if *Done = True:*
      $y_j = r_j$
    else:
      $y_j = r_j + \gamma \cdot max_a\left(\theta'(s_{j+1})\right)$
    end if
    Perform SGD on Huber_Loss$\left(y_j, \theta(s_j)\right)$ with respect network θ
    every ω *step update* $\theta' \leftarrow \theta$
    Update $s_t \leftarrow s_{t+1}$
  end if
  end for
end for

ALGORITMO A2C CON MUESTREO SENCILLO.

Tabla 8. Hiperparámetros A2C

Parámetro	***Valor***
1: Optimizador	*SGD*
2: Máximo número de pasos	*1 o 37*
3: Tasa de aprendizaje	*N.A*
4: Descuento(γ)	*0.9*
5. Tamaño Replay Buffer ($\mathcal{D}$)	*1 o 37*

6: Número de capas ocultas actor	*2*
7: Número de unidades por capa actor	*256*
8: Número salidas actor	*56, media y std por cada central*
9: Número de capas ocultas crítico	*2*
10: Número de unidades por capa crítico	*128*
11: Número salidas critico	*28*
12: Tasa aprendizaje red neuronal actor	*0.0025*
13: Tasa aprendizaje red neuronal crítico	*0.01*
14: Número de muestras por lote	*1 o 37*
15: Función de activación entrada	*RELU*
16: Función de activación capas ocultas	*ELU*
17: exploración-explotación	*Muestreo Gaussiano ejecutado por el actor*

Tabla 9. Algoritmo A2C para resolver el despacho energético económico, actualizando la política en cada paso

Algoritmo 4 *Algoritmo A2C single sample*

Entrada: *estado, acción.*
Parámetros: *parámetro de exploración $\in$, parámetro de aprendizaje.*
Salida: *política óptima.*

Get Economic Dispatch parameters
Initialize replay memory $\mathcal{D}$ to capacity N
Initalize Economic Dispathc model
Initialize Actor π_θ
Initialize Critic V_ϕ^π

for *each episode I* ***do****:*
  Initialize state s_0
  for *t=1....T_I in episode I* ***do:***
    sample action $a_t \sim \pi(a|\mu,\sigma) = \mathcal{N}(a|\mu,\sigma)$ according to current policy
    execute action a_t and observe reward r_t and state s_{t+1}
    set TD target $y_t = r + \gamma V_\phi^\pi(s_{t+1})$
    Update *critic minimizing loss $\delta_t = \left(y_t - V_\phi^\pi(s_t)\right)^2$*
    Update *actor policy minimizing loss:*

$$\mathcal{L} = -log\left(\mathcal{N}\left(a\middle|\mu(s_t),\sigma(s_t)\right)\right)\cdot\delta_t$$

    Update *$s_t \leftarrow s_{t+1}$*
  end for
end for

ALGORITMO A2C CON MUESTREO.

Tabla 10. Algoritmo A2C para resolver el despacho energético económico, actualizando la política en al final del periodo.

Algoritmo 4 *Algoritmo A2C multi-sample*

Entrada: *estado, acción.*
Parámetros: *parámetro de exploración $\in$, parámetro de aprendizaje.*
Salida: *política óptima.*

Get Economic Dispatch parameters
Initialize replay memory $\mathcal{D}$ to capacity N
Initalize Economic Dispathc model
Initialize Actor π_θ
Initialize Critic V_ϕ^π

for *each episode I* ***do****:*
  Initialize state s_0
  for *t=1….T_I in episode I* ***do:***
    sample action $a_t \sim \pi(a|\mu,\sigma) = \mathcal{N}(a|\mu,\sigma)$ according to current policy
    execute action a_t and observe reward r_t and state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  if *size batch < $\mathcal{D}$ size:*
    Sample all data (s_j, a_j, r_j, s_{j+1}) from $\mathcal{D}$
    set TD target $y_t = r + \gamma V_\phi^\pi(s_{t+1})$ from sample
    Update *critic minimizing loss $\delta_t = \left(y_t - V_\phi^\pi(s_t)\right)^2$*
    Update *actor policy minimizing loss:*

$$\mathcal{L} = -log\left(\mathcal{N}\left(a\middle|\mu(s_t),\sigma(s_t)\right)\right)\cdot\delta_t$$

    Update *$s_t \leftarrow s_{t+1}$*
    Clear *$\mathcal{D}$*
  end if
  end for
end for