



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Obras Civiles

**Optimización del Diseño de Soportes de
Transformadores Eléctricos Mediante el uso
de un Método de Aprendizaje por Refuerzo
Integrado con Software de Elementos
Finitos**

Presentada por:

Nicole Alejandra Cañipa Aedo

como requisito parcial para optar al título de la carrera de

Ingeniería Civil

Profesor Guía:

Danko José Jerez Urquieta

Enero 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía: Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Optimización del Diseño de Soportes de Transformadores Eléctricos Mediante el uso de un Método de Aprendizaje por Refuerzo Integrado con Software de Elementos Finitos

Nombre del candidato(a): Nicole Alejandra Cañipa Aedo

Carrera / Grado: Ingeniería Civil

Campus: Casa Central Valparaíso; **Departamento:** Obras Civiles

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, **Danko Jerez Urquieta**, en mi calidad de profesor guía del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis: Danko Jerez Urquieta

Fecha: 28.01.2026 ; **Firma:** 

Estudiante o Candidato(a): Nicole Cañipa Aedo

Fecha: 28.01.2026 ; **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Dedico este trabajo a mis padres,

Hernan Cañipa y Astrid Aedo

*por su amor incondicional, su apoyo constante
y por darme las herramientas para llegar hasta aquí.*

Índice

1. Introducción	6
1.1. Objetivos	8
1.2. Estructura de la Memoria	8
2. Marco Teórico	9
2.1. Diseño Estructural de Soportes en Chile	9
2.1.1. Tipología Estructural: Diagonales y Cantoneiras	9
2.1.2. Marco Normativo y Jerarquía Técnica	10
2.1.3. Cargas y Solicitaciones de Diseño	11
2.1.4. Combinaciones de Carga a Evaluar	12
2.1.5. Criterio de Estructura Rígida y Desacoplamiento Dinámico	14
2.1.6. Verificación de Elementos Finitos y Estabilidad	15
2.2. Fundamentos de Optimización Estructural	15
2.3. Aprendizaje por Refuerzo Profundo (DRL)	17
2.3.1. Deep Q-Network (DQN)	18
2.3.2. Estrategia de Eficiencia: Detención Temprana (Early Stopping)	20
3. Metodología	21
3.1. Formulación Matemática de la Optimización	22
3.2. Diseño del Entorno de Aprendizaje	23
3.2.1. Espacio de Estados	23
3.2.2. Espacio de Acciones	24
3.3. Función de Recompensa	24
3.4. Arquitectura del Sistema Computacional	25
3.4.1. Configuración del Agente y Red Neuronal	26

4. Ejemplo Ilustrativo	28
4.1. Definición del Espacio de Búsqueda y Materiales	29
4.2. Cargas y Criterios de Diseño	30
4.3. Solución Exacta: Método de Fuerza Bruta	30
4.4. Desempeño del Algoritmo DQN	32
4.4.1. Definición de Escenarios Experimentales	32
4.4.2. Entorno Computacional	33
4.4.3. Resultados de una Ejecución Típica	34
4.4.4. Interpretación de la Dinámica de Aprendizaje	34
4.5. Resultados Estadísticos y Dinámica de Aprendizaje	35
4.5.1. Análisis de la Velocidad de Convergencia	37
5. Ejemplo Aplicación	38
5.1. Análisis del Espacio de Búsqueda 3D	39
5.2. Resultados Estadísticos y Desempeño del Agente 3D	41
6. Discusión Comparativa: 2D vs 3D	42
6.1. Contracción del Espacio de Soluciones	42
6.2. Evolución de la Dificultad de Convergencia	43
7. Conclusiones	43
A. Anexos: Algoritmo de Optimización	45
A.1. Entorno Virtual y su Interacción con SAP2000	45
A.2. Agente DQN	46
B. Diagrama de Flujo del Algoritmo	48
C. Catálogo de Perfiles Estructurales	50

Resumen

El diseño de estructuras de soporte para equipos eléctricos de alta tensión en Chile enfrenta desafíos únicos debido a la alta sismicidad del país y a los estrictos requisitos de operación continua. La normativa vigente impone restricciones severas, destacando la necesidad de que estas estructuras sean rígidas, con una frecuencia fundamental superior a 30 Hz, para evitar la amplificación dinámica. El proceso de diseño tradicional, basado en métodos iterativos manuales y la experiencia del ingeniero, suele resultar costoso en tiempo y no garantiza el uso óptimo de los recursos disponibles.

Esta investigación presenta una metodología de optimización del diseño estructural basada en Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning, DRL). Se implementa un agente inteligente utilizando el algoritmo Deep Q-Network (DQN), el cual interactúa autónomamente con el software de elementos finitos SAP2000 mediante una interfaz de programación (API). El problema de diseño se formula como una optimización combinatoria en un espacio discreto, donde el agente aprende a seleccionar perfiles de acero de un catálogo comercial para minimizar el peso de la estructura. Para considerar las restricciones del método numérico, los criterios de resistencia y frecuencia fundamental se incorporan dentro de la función de recompensa; es decir, el algoritmo recibe una penalización matemática cada vez que propone un diseño que no cumple con la norma.

La metodología se valida en dos etapas: un modelo simplificado 2D y una torre reticulada espacial 3D sujeta a torsión y sismo bidireccional. Los resultados sugieren que el agente es capaz de converger al óptimo global, explorando el conjunto óptimo de manera robusta y reduciendo el tiempo de cómputo considerablemente con respecto a métodos de fuerza bruta. Finalmente, el estudio concluye que es fundamental permitir que la IA explore una gran variedad de soluciones al inicio para encontrar el diseño más eficiente. Esto confirma que la Inteligencia Artificial es una herramienta capaz de resolver problemas de ingeniería complejos con reglas difíciles de cumplir.

Palabras claves: Optimización estructural, Aprendizaje por Refuerzo, Deep Q-Network, SAP2000, Estructuras de acero, Diseño sísmico.

Abstract

The design of support structures for high-voltage electrical equipment in Chile faces unique challenges due to the country's high seismicity and strict continuous operation requirements. Current regulations impose severe restrictions, particularly highlighting the need for these structures to be rigid, with a fundamental frequency greater than 30 Hz, to prevent dynamic amplification. The traditional design process, based on manual iterative methods and engineering experience, is often time-consuming and does not guarantee the optimal use of available resources. This research presents a structural design optimization methodology based on Deep Reinforcement Learning (DRL). An intelligent agent is implemented using the Deep Q-Network (DQN) algorithm, which interacts autonomously with the finite element software SAP2000 via an Application Programming Interface (API). The design problem is formulated as a combinatorial optimization task within a discrete space, where the agent learns to select steel profiles from a commercial catalog to minimize the structure's weight. To account for the numerical method constraints, strength and fundamental frequency criteria are incorporated directly into the reward function; that is, the algorithm receives a mathematical penalty whenever it proposes a design that fails to comply with the standard. The methodology is validated in two stages: a simplified 2D model and a 3D spatial lattice tower subjected to torsion and bidirectional seismic loading. The results suggest that the agent is capable of converging to the global optimum, robustly exploring the solution space and significantly reducing computation time compared to brute force methods. Finally, the study concludes that allowing the AI to explore a wide variety of solutions in the early stages is fundamental to finding the most efficient design. This confirms that Artificial Intelligence is a capable tool for solving complex engineering problems governed by stringent constraints.

1. Introducción

El diseño de estructuras de acero se define por dos requisitos fundamentales: la necesidad de garantizar la máxima seguridad y resiliencia ante eventos extremos junto a la creciente necesidad de optimizar el uso de recursos con tal de asegurar la sostenibilidad económica y medioambiental de los proyectos. En este contexto, la optimización estructural surge tanto como una disciplina académica como una herramienta fundamental para la ingeniería moderna. El objetivo es encontrar, de manera sistemática, el diseño más eficiente para una estructura sujeta a un conjunto de restricciones de desempeño. Este proceso se plantea generalmente como la minimización de una función de costo (como el peso), apoyada en simulaciones numéricas, donde las variables de diseño (como la sección transversal de los perfiles) controlan la configuración estructural y son ajustadas iterativamente por un algoritmo (Belegundu y Chandrupatla, 2011).

Esta problemática es especialmente relevante en el diseño de estructuras de soporte para equipos eléctricos de alta tensión en países de alta sismicidad como Chile . Estos componentes son críticos para la infraestructura energética y se clasifican como “Equipos Semi-Rígidos”, por lo que deben cumplir estrictas normas que aseguren no solo que la estructura no colapse, sino que el equipo eléctrico pueda seguir operando inmediatamente después de un terremoto (Comité Chileno de CIGRE, 2020). Esto remarca la importancia de la normativa sísmica nacional, la cual impone estrictas restricciones de resistencia, rigidez y estabilidad (Comisión Nacional de Energía, 2019).

Tradicionalmente, el diseño estructural es un proceso iterativo que depende en gran medida de la intuición del ingeniero (Dworschak et al., 2022). El profesional suele elegir una configuración inicial basada en su experiencia y luego realiza múltiples ciclos de cálculo y ajuste manual utilizando software de ingeniería (CAE/FEM). Este enfoque de “prueba y error” es lento y consume mucho tiempo, ya que requiere validar constantemente parámetros complejos como frecuencias naturales y esfuerzos (Dworschak et al., 2022; Fu et al., 2024), sin embargo, este enfoque no siempre garantiza encontrar soluciones óptimas. El problema puede formularse matemáticamente como una optimización combinatoria no convexa (Millan y Filho, 2019): es combinatoria porque las

variables (los perfiles de acero) se seleccionan de un catálogo discreto de secciones comerciales, y es no convexa porque el espacio de soluciones contiene múltiples óptimos locales, dificultando la localización del óptimo global (Sutton y Barto, 2018).

La motivación principal de este trabajo surge de la necesidad de modernizar las prácticas de la ingeniería estructural en Chile, integrando herramientas computacionales avanzadas que permitan superar las limitaciones del diseño manual. En un escenario donde los plazos de los proyectos son cada vez más acotados y las exigencias de eficiencia material son mayores, desarrollar un sistema capaz de explorar miles de alternativas de diseño de forma autónoma no solo reduce el error humano, sino que libera al ingeniero de tareas repetitivas, permitiéndole enfocarse en decisiones de mayor valor estratégico.

La presente investigación aborda este desafío explorando el potencial del aprendizaje automático para automatizar y mejorar el proceso de diseño (Sun et al., 2020). Específicamente, se propone una metodología innovadora que integra Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning, DRL) con software de análisis por elementos finitos (SAP2000) para encontrar diseños livianos que satisfagan todas las exigencias normativas (Comité Chileno de CIGRE, 2020). El DRL surge como una opción viable para el diseño automatizado, ya que minimiza la dependencia de grandes cantidades de datos y mejora la integración física en el proceso de automatización (Fu et al., 2024). Mediante frameworks de DRL, un agente de IA aprende de forma autónoma a tomar decisiones de diseño eficientes y precisas. El Deep Q-Network (DQN), como metodología de DRL, se utiliza para aproximar la función Q con el objetivo de maximizar las recompensas acumuladas (Hu et al., 2019). Al integrar este agente DQN con la capacidad de análisis estructural de elementos finitos de SAP2000, se busca automatizar las tareas repetitivas y complejas (Dworschak et al., 2022), permitiendo una optimización que va más allá de la mera factibilidad (Fu et al., 2024).

La relevancia de este trabajo radica en su potencial para generar diseños más eficientes en términos de material y costo (Millan y Filho, 2019) y, fundamentalmente, validar la aplicación de técnicas de inteligencia artificial de vanguardia a un problema de ingeniería civil real. Basado en esta oportunidad de automatización, la presente memoria de título se guía por la siguiente hipótesis: “La

implementación de un agente de Aprendizaje por Refuerzo Profundo (DQN) integrado con análisis de elementos finitos permite obtener diseños estructurales optimizados que cumplen la normativa chilena, superando en eficiencia al diseño manual tradicional.”

1.1. Objetivos

El objetivo general es: Desarrollar e implementar un sistema de optimización estructural autónomo.

Los objetivos específicos son:

- Desarrollo e implementación de un agente de optimización basado en Deep Q Network.
- Entrenamiento, validación y análisis comparativo del desempeño del agente de optimización.

1.2. Estructura de la Memoria

El presente documento se organiza en ocho capítulos que abordan secuencialmente el desarrollo de la investigación. Tras presentar el resumen y la introducción a la problemática, el Capítulo 3 establece las bases teóricas sobre la normativa sísmica chilena y los algoritmos de Aprendizaje por Refuerzo Profundo. El Capítulo 4 detalla la metodología propuesta, incluyendo la formulación matemática de la optimización y la arquitectura computacional que integra el agente inteligente con SAP2000. La validación experimental se desarrolla en dos etapas progresivas: el Capítulo 5 presenta un modelo ilustrativo en 2D para calibrar los hiperparámetros, mientras que el Capítulo 6 aborda la aplicación final sobre una torre reticulada espacial en 3D. Posteriormente, el Capítulo 7 ofrece una discusión crítica y comparativa sobre el comportamiento del algoritmo y la física del problema entre ambos entornos, finalizando con las conclusiones principales y líneas de trabajo futuro en el Capítulo 8.

2. Marco Teórico

La presente investigación se fundamenta en la intersección de tres disciplinas: la ingeniería estructural sísmica, la optimización matemática y la inteligencia artificial. A continuación, se detallan los principios teóricos y normativos que sustentan el desarrollo de la metodología propuesta.

2.1. Diseño Estructural de Soportes en Chile

El diseño de estructuras de soporte para equipos eléctricos de alta tensión en Chile se desarrolla bajo un marco de exigencias técnicas singulares, derivadas de la elevada amenaza sísmica del territorio y la criticidad de la infraestructura energética para la seguridad del país (Comisión Nacional de Energía, 2019). Estas estructuras, tipificadas comúnmente como estructuras bajas o reticulados de acero, deben garantizar no solo la integridad física de sus elementos frente a cargas extremas, sino también la inmediata disponibilidad de operación del equipo soportado (transformadores, desconectores, pararrayos, entre otros) tras un evento sísmico severo. Bajo este criterio, cualquier daño estructural no debe intervenir con el funcionamiento de la red ni comprometer la continuidad del servicio.

2.1.1. Tipología Estructural: Diagonales y Cantoneras

Las estructuras de soporte tipo reticulado se componen principalmente de dos grupos de elementos:

1. **Cantoneras:** Son los elementos verticales principales ubicados en las esquinas de la torre. Su función principal es resistir las cargas axiales (compresión y tracción) derivadas del peso propio y del momento de volcamiento sísmico.
2. **Diagonales:** Son elementos inclinados que conectan las cantoneras, formando triangulaciones. Su rol es proveer rigidez lateral a la estructura y transmitir el corte sísmico hacia las fundaciones.

La Figura 1 ilustra la disposición típica de estos elementos en una torre de soporte estándar.

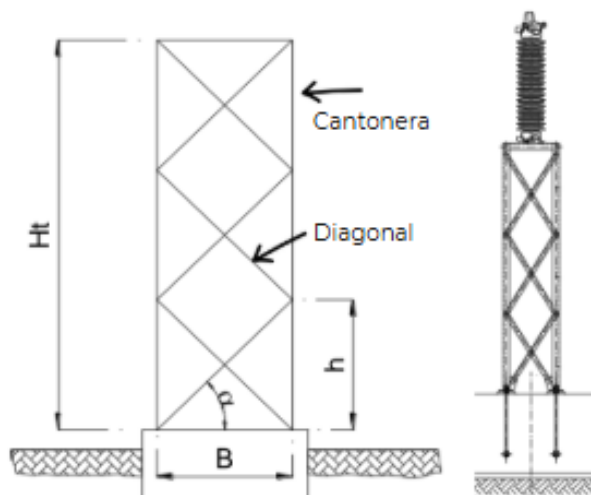


Figura 1: Esquema de una estructura de soporte reticulada, identificando Cantoneras y Diagonales.

Donde:

- H_t : Altura total de la torre.
- h : Altura de cada nivel de diagonales.
- B : Ancho basal.

El proceso de diseño se rige fundamentalmente por el método de Diseño por Factores de Carga y Resistencia (LRFD), conforme a lo estipulado en el Anexo Técnico de Requisitos Sísmicos vigente (Comité Chileno de CIGRE, 2020). A continuación, se detallan las consideraciones normativas, las solicitaciones actuantes y los criterios de aceptación (incluyendo límites de resistencia, rigidez y desplazamientos elásticos) que definen el espacio de soluciones factibles para el problema de optimización estructural (Transelec S.A., 2013).

2.1.2. Marco Normativo y Jerarquía Técnica

La validación estructural se fundamenta en una jerarquía normativa que abarca desde estándares nacionales generales hasta especificaciones técnicas particulares de la industria eléctrica. Los documentos que rigen el diseño y cálculo estructural para esta investigación son:

- **NCh2369.Of2023 (Instituto Nacional de Normalización, 2023)**: “Diseño sísmico de estructuras e instalaciones industriales”. Establece los criterios generales para el análisis sísmico en el ámbito industrial chileno.
- **ETG-A.0.20**: Especificación técnica de Transelec S.A. para el diseño sísmico de equipos y soportes (Transelec S.A., 2013). Esta norma entrega criterios obligatorios para garantizar la inmediata disponibilidad de operación después de un sismo en la transmisión en alta tensión en Chile, imponiendo restricciones más severas que la norma general, especialmente en términos de rigidez y espectros de respuesta.
- **ASCE 10-15**: “Design of Latticed Steel Transmission Structures” (American Society of Civil Engineers, 2015). Se utiliza como referencia internacional para la verificación de la capacidad resistente de los miembros de acero (perfiles ángulos), definiendo las fórmulas para el cálculo de resistencia a compresión, tracción y flexión.

2.1.3. Cargas y Solicitaciones de Diseño

El modelo estructural debe ser sometido a un conjunto de cargas que simulan las condiciones de servicio y eventos extremos. Según el Anexo Técnico de Requisitos Sísmicos (Comisión Nacional de Energía, 2019), estas se clasifican en:

1. **Cargas Permanentes (CP)**: Incluyen el peso propio de la estructura de acero y del equipo eléctrico (especificado por el fabricante). Dada la masa de los transformadores, su ubicación genera un efecto de péndulo invertido crítico para el análisis dinámico (Comité Chileno de CIGRE, 2020).
2. **Solicitaciones Sísmicas (E)**: Representan la demanda dominante. Se modelan considerando una aceleración basal efectiva mas desfavorable ($A_0 = 0,5g$) según las especificaciones técnicas (Transelec S.A., 2013) con tal de asegurar la disponibilidad inmediata tras un sismo.

Para estructuras reticuladas de acero, se adopta un factor de modificación de respuesta $R = 3$, limitando el diseño mayormente al rango elástico (Comisión Nacional de Energía, 2019).

3. **Cargas de Operación (CO / COs):** Derivadas del funcionamiento normal, como presiones internas y vibraciones. Se distinguen las de diseño (CO) y las simultáneas con sismo (COs) (CNE2025).
4. **Cargas de Conexión (T):** Efectos dinámicos de “Tirón” en los terminales. Para equipos de tensión mayor a 245 kV, se considera una carga estática equivalente de 1.75 [kN] (175 [daN]) (Comisión Nacional de Energía, 2019).
5. **Cargas de Cortocircuito (F_c):** Fuerzas electrodinámicas calculadas con la corriente dinámica de cortocircuito. Aunque son eventos de baja probabilidad, generan momentos significativos en la base Comisión Nacional de Energía, 2019.
6. **Condiciones Meteorológicas (CA / CAs):** Cargas de viento, nieve o hielo según la zona geográfica Comisión Nacional de Energía, 2019.

En el contexto específico de esta investigación, se establece una jerarquización de las sollicitaciones. Dado que los equipos de alta tensión poseen una masa elevada concentrada en la parte superior de la estructura metálica, las fuerzas inerciales sísmicas (E) tienden a ser significativamente mayores que las fuerzas aerodinámicas del viento (CA). Por consiguiente, para efectos del entrenamiento del agente DQN y para evitar un costo computacional excesivo en la evaluación de miles de estados, el diseño estará gobernado por la interacción Sismo-Peso, desestimando las cargas meteorológicas (CA/CAs) bajo la hipótesis de que una estructura capaz de resistir la demanda sísmica de 0,5g también satisface las demandas de viento.

2.1.4. Combinaciones de Carga a Evaluar

El algoritmo de diseño debe verificar la resistencia de la estructura bajo el método LRFD (Load and Resistance Factor Design), evaluando los Estados Límites Últimos. De acuerdo con

el Artículo 32 del Anexo Técnico Comisión Nacional de Energía, 2019, se definen las siguientes combinaciones que determinan la resistencia requerida última (U) que el agente debe satisfacer:

Caso No Sísmico (Gravitacional y Operacional): Verifica la capacidad de la estructura para soportar cargas estáticas y eventuales fallas eléctricas sin sismo.

$$CU1 = 1,2CP + 1,2CO + 1,2Fc + 1,6CA \quad (1)$$

Casos Sísmicos: Consideran la acción bidireccional del sismo ($E = \pm E_h \pm E_v$), donde E_h y E_v representan las componentes horizontal y vertical de la carga sísmica, respectivamente. Estas son las combinaciones críticas que gobiernan la optimización del agente:

$$CU2 = 1,2CP + 1,4E + 1,2CO_s + 1,2T + 1,2(0,6Fc) \quad (2)$$

$$CU3 = 1,2CP + 1,4E + 1,2CO_s + 1,2T + 1,2(0,6Fc) + 1,6CAs \quad (3)$$

Adicionalmente, se deben verificar los casos de descarga gravitacional para asegurar la estabilidad global (evitar el levantamiento de fundaciones o vuelco de la estructura) cuando el sismo actúa en sentido opuesto a la gravedad:

$$CU4 = 1,0CP + 1,4E + 1,2CO_s + 1,2T + 1,2(0,6Fc) \quad (4)$$

$$CU5 = 1,0CP + 1,4E + 1,2CO_s + 1,2T + 1,2(0,6Fc) + 1,6CAs \quad (5)$$

El agente de Aprendizaje por Refuerzo será penalizado si el Factor de Utilización (FU) de cualquier elemento supera la unidad ($FU > 1,0$) bajo la envolvente de estas combinaciones.

2.1.5. Criterio de Estructura Rígida y Desacoplamiento Dinámico

Una de las restricciones más determinantes en el diseño de soportes para equipos de alta tensiones la exigencia de rigidez o frecuencia fundamental. La normativa chilena, a través del Anexo Técnico de Requisitos Sísmicos y especificaciones industriales como la ETG-A.0.20, clasifica como Sistemas Estructurales Rígidos a aquellos cuya frecuencia fundamental de oscilación (f_n) es mayor o igual a 30 [Hz] (Comisión Nacional de Energía, 2019).

$$f_n \geq 30 \text{ [Hz]} \quad (6)$$

El objetivo de esta restricción es doble:

- **Desacoplamiento Dinámico:** Busca evitar fenómenos de resonancia entre la estructura de soporte y las frecuencias predominantes del evento sísmico. Esto garantiza que la estructura no genere una amplificación dinámica de las aceleraciones del suelo que se transmiten al equipo, el cual suele poseer componentes de material frágil como la porcelana (Comisión Nacional de Energía, 2019).
- **Simplificación del Análisis:** Al garantizar que la estructura se comporte como un cuerpo rígido, se permite realizar verificaciones sísmicas utilizando directamente la aceleración máxima del suelo (A_0), simplificando el cálculo al no requerir factores de amplificación dinámica.

Para el algoritmo de optimización (DQN), esta restricción de frecuencia actúa como una barrera no lineal severa (Fu et al., 2024). En la práctica, el objetivo del agente es minimizar el uso de material (peso); sin embargo, reducir la sección transversal de los perfiles tiende a disminuir la rigidez global, provocando que el diseño caiga por debajo del límite de los 30 Hz y se convierta en un sistema flexible, lo cual es penalizado por la función de recompensa al no cumplir con el criterio de diseño original (Fu et al., 2024).

2.1.6. Verificación de Elementos Finitos y Estabilidad

La validación del diseño se realiza mediante la verificación individual de cada componente. Para una estructura reticulada compuesta por perfiles angulares (L), se evalúan los estados límites de falla y fenómenos de inestabilidad bajo los siguientes criterios:

1. **Factor de Utilización (FU):** Relación entre la demanda última (P_u) y la capacidad nominal reducida (ϕP_n). Para todo miembro i , se debe cumplir:

$$FU_i = \frac{P_{u,i}}{\phi P_{n,i}} \leq 1,0 \quad (7)$$

La capacidad ϕP_n se calcula según ASCE 10-15, considerando los efectos de pandeo flexural en compresión y fluencia/rotura en tracción.

2. **Esbeltez Global (KL/r):** Para garantizar la estabilidad geométrica y evitar vibraciones excesivas durante el montaje o servicio, la normativa limita la relación de esbeltez.

$$\frac{KL}{r} \leq \lambda_{max} \quad (8)$$

Donde λ_{max} toma valores típicos de 150 para miembros principales en compresión y 200 para miembros en tracción o secundarios. En esta fórmula:

- K : Factor de longitud efectiva, que depende de las condiciones de empotramiento en los extremos del miembro.
- L : Longitud no arriostrada del elemento.
- r : Radio de giro de la sección transversal del perfil.

2.2. Fundamentos de Optimización Estructural

Desde una perspectiva matemática, el diseño óptimo de la estructura de soporte se plantea como un problema de Optimización No Lineal (NLP) y de carácter combinatorio no convexo

(Belegundu y Chandrupatla, 2011). El objetivo fundamental es encontrar soluciones que no solo sean factibles, sino óptimas, maximizando la rentabilidad y el rendimiento de los recursos, En el caso de estructuras de acero, esto se traduce en minimizar el costo material, representado por el peso total de la estructura (W), asegurando simultáneamente el cumplimiento de todas las exigencias normativas de resistencia, rigidez y estabilidad (Fu et al., 2024; Millan y Filho, 2019).

Formalmente, el problema se define como la búsqueda de un vector de variables de diseño x que satisfaga:

$$\begin{aligned} & \underset{p_c, p_d \in C}{\text{mín}} && W(p_c, p_d) \\ \text{sujeto a:} &&& f(p_c, p_d) \geq f_{min} \end{aligned} \quad (9)$$

Verificaciones LRFD (Resistencia y Estabilidad)

Donde:

- p_c, p_d : Son las variables de diseño discretas (índices del catálogo) correspondientes a los grupos de elementos de Cantoneras y Diagonales, respectivamente.
- C : Representa el catálogo discreto de perfiles comerciales disponibles (e.g., perfiles ángulo L según norma).
- $W(p_c, p_d)$: Es la función objetivo que representa el peso total de la estructura, dependiente de la geometría y densidad del acero.
- $f(p_c, p_d)$: Es la frecuencia natural fundamental de la estructura, la cual debe superar el umbral $f_{min} = 30$ Hz para cumplir con la condición de estructura rígida.

Esta formulación, aunque visualmente compacta, encierra una alta complejidad computacional caracterizada por dos factores principales:

1. **Explosión Combinatoria:** Al ser C un conjunto discreto y finito, no es posible utilizar derivadas o gradientes para encontrar el mínimo. Si el catálogo C tiene M perfiles y la estructura se

divide en N grupos de diseño, el espacio de búsqueda total es M^N . Para un catálogo estándar de 40 perfiles y solo 2 grupos de variables (p_c, p_d) , existen 1,600 combinaciones, pero al aumentar la dimensionalidad del diseño, este número se vuelve inmanejable para métodos de fuerza bruta (Belegundu y Chandrupatla, 2011).

2. **No Convexidad y Restricciones “Caja Negra”:** La función de frecuencia $f(p_c, p_d)$ y las verificaciones LRFD no son lineales ni convexas. Un pequeño cambio en el perfil de la diagonal puede alterar drásticamente la rigidez global o los modos de vibración, generando múltiples óptimos locales. Además, evaluar si una combinación cumple con las restricciones requiere ejecutar una simulación completa de Elementos Finitos (SAP2000) en cada iteración, lo que impide tener una expresión analítica cerrada del problema (Millan y Filho, 2019).

Debido a esto, se justifica el uso de un agente de Aprendizaje por Refuerzo, capaz de navegar este espacio discreto y discontinuo para encontrar la combinación (p_c, p_d) que minimice W sin violar f_{min} ni los criterios de seguridad.

2.3. Aprendizaje por Refuerzo Profundo (DRL)

El Aprendizaje por Refuerzo (RL, por sus siglas en inglés) es un paradigma del aprendizaje automático, derivado del Machine Learning (ML) donde un agente aprende a tomar decisiones secuenciales interactuando con un entorno dinámico para maximizar una señal de recompensa acumulada (Hansen, 2016). A diferencia del aprendizaje supervisado, no se parte de datos etiquetados, sino que el conocimiento se genera a través de la experiencia directa de prueba y error (Sun et al., 2020).

Para formalizar el diseño estructural como un problema de aprendizaje secuencial, se utiliza el Proceso de Decisión de Markov (MDP). Este marco matemático permite que el agente aprenda a través de la interacción continua con un entorno de simulación (Fu et al., 2024; Hu et al., 2019). En este contexto, la tupla (S, A, R, γ) se define de la siguiente manera:

- Espacio de Estados (S): Representa la descripción numérica detallada de la estructura en un momento dado t (Fu et al., 2024). En el diseño de soportes, el estado incluye las dimensiones actuales de los perfiles, las propiedades de los materiales, las cargas aplicadas y, fundamentalmente, los resultados de las verificaciones normativas (como factores de utilización de esfuerzos y desplazamientos elásticos). Esto permite al agente evaluar qué tan “segura” o “ineficiente” es la estructura antes de realizar un cambio.
- Espacio de Acciones (A): El conjunto de decisiones que el agente puede tomar para modificar el estado. En este caso, las acciones consisten en seleccionar perfiles específicos de un catálogo comercial para asignarlos a los diferentes grupos de miembros de la estructura. Estas acciones son de naturaleza discreta y combinatoria, permitiendo al agente explorar diversas configuraciones geométricas de manera sistemática (Dworschak et al., 2022).
- Función de Recompensa (R): Es la señal de retroalimentación que indica al agente si su acción fue acertada en función de los objetivos de diseño (Hu et al., 2019). El agente aprende a maximizar esta señal acumulada para hallar el equilibrio entre seguridad y economía (Wei et al., 2020), penalizando severamente el incumplimiento normativo (diseños infactibles por $FU > 1,0$ o $f < 30$ Hz) y premiando proporcionalmente la reducción de peso solo cuando el diseño es seguro.
- Factor de Descuento (γ): Es un parámetro que determina la importancia de las recompensas futuras frente a las inmediatas (Dworschak et al., 2022). Un γ cercano a 1 (por ejemplo, 0.99) obliga al agente a tener una “visión a largo plazo”, entendiendo que una secuencia de cambios pequeños puede llevar a una estructura optimizada globalmente, aunque las recompensas individuales en los pasos intermedios sean bajas (Sun et al., 2020).

2.3.1. Deep Q-Network (DQN)

En problemas simples, se utiliza una tabla (Q-table) para almacenar el valor de cada acción en cada estado. Sin embargo, el diseño estructural enfrenta la “maldición de la dimensionalidad”,

donde las combinaciones posibles de perfiles y geometrías pueden hacer inviable el uso de tablas (Dworschak et al., 2022). Esto se debe ya que a medida que aumenta el número de variables de estado (o si el espacio de estados es continuo), el número total de estados crece exponencialmente. Esto requiere una cantidad de memoria, además no se trata solo de un problema de almacenamiento, sino también del tiempo y los datos necesarios para llenar la tabla con precisión (Sutton y Barto, 2018).

DQN soluciona esto integrando Redes Neuronales Profundas (DNN) como aproximadores de funciones. La red neuronal toma el estado s como entrada y predice el Valor-Q ($Q(s,a, \theta)$), que representa la utilidad esperada de tomar la acción a en dicho estado. El objetivo del entrenamiento es ajustar los parámetros de la red (θ) para que estas predicciones sean lo más precisas posible según la Ecuación de Bellman que se presenta a continuación (Hansen, 2016).

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q_{target}(s', a'; \theta^-) - Q(s, a) \right] \quad (10)$$

Donde:

- α : Es la tasa de aprendizaje (learning rate), que determina la velocidad y estabilidad de la convergencia del modelo y sirve para medir la influencia de la información nueva sobre los conocimientos adquiridos por el agente (Dworschak et al., 2022), se tiene $\alpha \in [0,1]$.
- s' : Representa el estado siguiente resultante de ejecutar la acción a .
- a' : Representa las posibles acciones futuras en el estado s' .
- θ^- : Son los parámetros (pesos y sesgos) de la Red Objetivo (Target Network) (Hu et al., 2019). Se usa para calcular el valor Q del siguiente estado s_{t+1} y ayuda a mitigar el problema de la sobreestimación de los valores de acción, proporcionando una base más estable para las actualizaciones de la red principal.

Para que un agente DQN aprenda de manera estable, se utilizan tres mecanismos:

- Memoria de Repetición (Experience Replay): El agente almacena sus experiencias pasadas en un búfer. Durante el entrenamiento, se extraen muestras aleatorias para actualizar la red, lo que rompe la correlación temporal de las acciones sucesivas y permite aprender de éxitos y fracasos previos de manera eficiente (Hu et al., 2019).
- Red Objetivo (Target Network): Para evitar que el objetivo de aprendizaje cambie constantemente (lo que causaría oscilaciones), se utiliza una segunda red neuronal con parámetros fijos temporalmente (θ^-) para calcular los valores objetivo. Esta red se sincroniza con la red principal solo cada cierto número de pasos (Dworschak et al., 2022).
- Política ϵ -greedy: Al inicio, el agente toma acciones aleatorias con alta probabilidad (ϵ) para explorar el espacio de soluciones. A medida que el entrenamiento progresa, ϵ disminuye y el agente empieza a explotar el conocimiento adquirido, eligiendo las acciones con mayor Valor-Q (Hansen, 2016).

2.3.2. Estrategia de Eficiencia: Detención Temprana (Early Stopping)

Un desafío crítico en la optimización estructural integrada con software comercial (como SAP2000) es que el proceso de diseño es notoriamente lento y consume mucho tiempo debido al alto costo computacional de cada evaluación de la función de recompensa, la cual actúa como una “caja negra” física que requiere un análisis FEM completo (Honda et al., 2022). Dado que el esfuerzo de simulación excede por mucho al esfuerzo de adaptación de la red neuronal, es vital evitar el gasto innecesario de recursos mediante un mecanismo de Detención Temprana (Dworschak et al., 2022).

Esta técnica monitorea el desempeño del agente utilizando una ventana deslizante de los últimos N episodios para evaluar indicadores de convergencia y esfuerzo de entrenamiento (Dworschak et al., 2022). El proceso de aprendizaje se detiene automáticamente si se alcanzan condiciones de estabilidad:

- Convergencia a una solución estable: El agente ha minimizado el error y la varianza en la

recompensa obtenida, indicando que la política se ha estabilizado (Hu et al., 2019; Wei et al., 2020).

- Estancamiento del rendimiento: No se observa una mejora significativa en la función objetivo (minimización del peso material) durante un periodo de “paciencia” predefinido (Hertel et al., 2020).

Esta estrategia asegura que el algoritmo finalice una vez que se ha encontrado una política de diseño óptima y robusta, haciendo viable la aplicación de la metodología en entornos de ingeniería reales donde la capacidad de cómputo es limitada (Fu et al., 2024). La Figura 2.2 ilustra la interacción cíclica implementada permite que el agente observe la configuración, decida cambios de perfil mediante su red neuronal, ejecute el análisis vía API para validar la física del cambio y reciba una recompensa basada en la eficiencia y el cumplimiento de los requisitos sísmicos normativos (Fu et al., 2024). Además se puede ver el diagrama de flujo del algoritmo en el Anexo B.

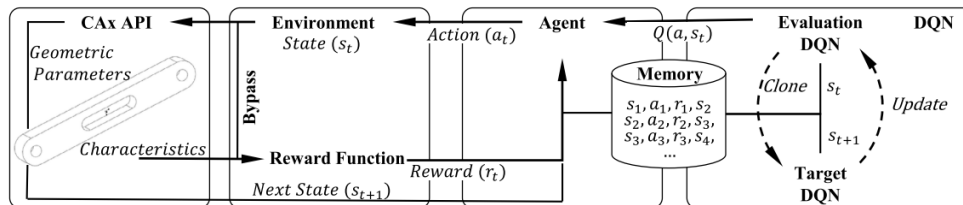


Figura 2.2: Esquema de funcionamiento del Algoritmo DQN aplicado al diseño estructural (Dworschak et al., 2022).

3. Metodología

Esta sección detalla la formulación del problema de ingeniería como un desafío de optimización matemática y su posterior traducción a un entorno de Aprendizaje por Refuerzo (RL). Se describe la arquitectura computacional desarrollada para integrar el agente inteligente con el motor de cálculo estructural y se definen los casos de estudio utilizados para la validación.

3.1. Formulación Matemática de la Optimización

El diseño óptimo de la estructura de soporte se plantea como un problema de optimización combinatoria multi-objetivo con restricciones no lineales. El objetivo es minimizar el peso total de la estructura de acero y su frecuencia natural sin incumplir con las restricciones de diseño, garantizando simultáneamente la integridad estructural (resistencia y estabilidad) y la operatividad del equipo eléctrico (rigidez dinámica).

Matemáticamente, el problema se formula de la siguiente manera:

$$\text{mín } W(\mathbf{x}) = \sum_{i=1}^{N_g} \rho \cdot A_i(x_i) \cdot L_i \quad (11)$$

Sujeto a:

$$g_1(\mathbf{x}) : \frac{P_{u,i}}{\phi P_{n,i}} \leq 1,0 \quad \forall i \in \text{Miembros} \quad (12)$$

$$g_2(\mathbf{x}) : f_1 \geq 30,0 \text{ [Hz]} \quad (13)$$

$$g_3(\mathbf{x}) : \frac{KL}{r} \leq \lambda_{max} \quad (14)$$

$$g_4(\mathbf{x}) : \mathbf{x} \in C \quad (15)$$

Donde:

- \mathbf{x} es el vector de variables de diseño, que contiene los índices de los perfiles asignados a cada grupo de elementos.
- N_g es el número de grupos de elementos estructurales (para este estudio: Cantoneras y Diagonales).
- ρ es la densidad del acero (7850 kg/m^3).
- $A_i(x_i)$ es el área de la sección transversal del grupo i , la cual depende del perfil seleccionado

por la variable x_i .

- L_i es la longitud total de los elementos pertenecientes al grupo i .
- La restricción (12) asegura que el Factor de Utilización (FU) no exceda la unidad para ningún elemento bajo las combinaciones de carga LRFD, según la norma ASCE 10-15.
- La restricción (13) impone que la frecuencia natural fundamental (f_1) sea superior a 30 Hz, cumpliendo con la especificación de estructura rígida de la ETG-A.0.20 Transelec S.A., 2013.
- La restricción (14) limita la esbeltez global para evitar problemas de vibración e inestabilidad.
- La restricción (15) indica que las variables son discretas y deben pertenecer al catálogo comercial disponible C (perfiles tipo L).

3.2. Diseño del Entorno de Aprendizaje

Para que el algoritmo DQN pueda resolver la formulación matemática anterior, el problema se modela como un MDP, según se discute en la Sección 2.3 .

3.2.1. Espacio de Estados

El estado s_t representa la configuración geométrica actual de la estructura en el paso de tiempo t . Dado que la topología es fija, el estado se define únicamente por las secciones transversales asignadas. El agente percibe el estado como un vector de enteros:

$$s_t = [idx_c, idx_d] \quad (16)$$

Donde cada índice apunta a una posición específica dentro del catálogo de perfiles ordenado por peso (de menor a mayor), asegurando que $idx = 0$ corresponda al perfil más liviano y $idx = N$ al más pesado. Esto permite al agente entender la “dirección” de la optimización: un índice mayor implica generalmente un perfil más robusto y pesado.

3.2.2. Espacio de Acciones

Se define un espacio de acciones discreto e incremental. En lugar de permitir que el agente elija cualquier perfil del catálogo en un solo paso (lo que generaría un espacio de acciones inmensamente grande), el agente realiza modificaciones graduales al diseño actual. El conjunto de acciones posibles \mathcal{A} es:

- a_0 : Aumentar perfil de Cantonera ($idx_c \leftarrow idx_c + 1$).
- a_1 : Disminuir perfil de Cantonera ($idx_c \leftarrow idx_c - 1$).
- a_2 : Aumentar perfil de Diagonal ($idx_d \leftarrow idx_d + 1$).
- a_3 : Disminuir perfil de Diagonal ($idx_d \leftarrow idx_d - 1$).

Este enfoque permite al agente “navegar” por el espacio de soluciones buscando el equilibrio óptimo localmente antes de explorar configuraciones drásticamente diferentes.

3.3. Función de Recompensa

Para garantizar la estabilidad numérica del entrenamiento, se implementó una función de recompensa acotada en el rango $[-1, 1]$. La recompensa R_t se define distinguiendo entre estados infactibles (que violan la normativa) y estados factibles.

$$R_t(w, f, r) = \begin{cases} -1,0 & \text{si el diseño es Infactible} \\ \beta \cdot S_w + (1 - \beta) \cdot S_f & \text{si el diseño es Factible} \end{cases} \quad (17)$$

Se considera un diseño Infactible, si la frecuencia fundamental es $f < 30$ Hz o el FU es $r > 1,0$.

Si el diseño es Factible, se calcula una suma ponderada (con $\beta = 0,7$) de dos puntuaciones simplificadas:

1. Puntuación de Peso (S_w): Incentiva linealmente la reducción de masa dentro de los límites del catálogo:

$$S_w = \frac{W_{max} - W_{actual}}{W_{max} - W_{min}} \quad (18)$$

2. Puntuación de Frecuencia (S_f): Se define una relación inversa simple. Dado que el objetivo es cumplir el límite inferior de 30 Hz sin sobredimensionar la rigidez:

$$S_f = \frac{30,0}{f_{actual}} \quad (19)$$

Esta fórmula asigna un valor máximo de 1,0 cuando la frecuencia está exactamente en el límite normativo (30 Hz). A medida que la frecuencia aumenta innecesariamente (ej. 60 Hz), el puntaje disminuye (0,5), penalizando el exceso de rigidez que no aporta valor al diseño óptimo. Cabe destacar que se pueden implementar diversas alternativas en la formulación de la función de recompensa.

3.4. Arquitectura del Sistema Computacional

La implementación tecnológica se basa en una arquitectura cliente-servidor, donde el agente de Inteligencia Artificial (Python) actúa como el cerebro que toma decisiones, y el software de elementos finitos (SAP2000) actúa como el entorno físico que evalúa las consecuencias.

La comunicación se realiza a través de la interfaz OAPI (Open Application Programming Interface) de CSI, utilizando la librería `comtypes` en Python. El flujo de trabajo cíclico, implementado en la clase `VirtualSapEnv` (Anexo A.1), sigue los siguientes pasos:

1. Inicialización: Python abre una instancia de SAP2000, define materiales (ASTM A36) y genera la geometría base del soporte.
2. Acción del Agente: La red neuronal DQN selecciona una acción (ej. "Disminuir Diagonal"). Python traduce esta acción actualizando las propiedades de la sección transversal en el modelo de SAP2000.

3. Análisis FEM: Se instruye a SAP2000 para ejecutar dos tipos de análisis:
 - Análisis Modal: Para extraer la frecuencia fundamental (f_1).
 - Análisis Estático Lineal: Con combinaciones de carga LRFD para obtener las fuerzas internas (P_u).
4. Extracción de Resultados: Python consulta los resultados a través de la API. Se recuperan los FU, las frecuencias naturales y el peso total de la estructura.
5. Cálculo de Recompensa: Con los datos extraídos, se evalúa la función de recompensa y se determina si el estado es terminal o si el episodio continúa.

Para mitigar fugas de memoria conocidas en la API de SAP2000 tras cientos de ejecuciones, se implementó una lógica de reinicio automático del software cada 50 iteraciones.

3.4.1. Configuración del Agente y Red Neuronal

El núcleo de toma de decisiones del agente DQN (Anexo A.2), implementa un Perceptrón Multicapa (MLP), también conocido en la literatura como Red Neuronal de Alimentación Hacia Adelante (Feed-Forward Neural Network). Esta arquitectura actúa como un aproximador de funciones universal, permitiendo estimar la función de valor de acción $Q(s, a; \theta)$ en un espacio de estados complejo donde el uso de tablas convencionales (Q-Tables) sería inviable (Dworschak et al., 2022).

El MLP diseñado consiste en una estructura de capas secuenciales donde la información fluye únicamente hacia adelante (desde la entrada hacia la salida). Sus características principales son:

- **Capas de Neuronas:** La red está compuesta por una capa de entrada, dos capas ocultas y una capa de salida.
- **Conectividad Total (Fully Connected):** La red es densa, lo que significa que cada neurona de una capa está vinculada matemáticamente a todas las neuronas de la capa subsiguiente mediante pesos sinápticos y sesgos ajustables (Dworschak et al., 2022).

- **No Linealidad:** Cada unidad en las capas ocultas aplica una función de activación no lineal a la suma ponderada de sus entradas. En este trabajo se utilizó la función **ReLU (Rectified Linear Unit)** (Sun et al., 2020), definida como $f(x) = \max(0, x)$, lo que permite al modelo aprender relaciones complejas y fronteras de decisión no lineales entre los estados factibles e infactibles.

Dentro del flujo de trabajo, la red neuronal cumple el rol de mapear la observación del entorno a una predicción de utilidad futura.

- **Entrada (Input Layer):** Recibe un vector que representa el estado del entorno s_t . Para este modelo, la entrada tiene una dimensión de 2 neuronas, correspondientes a los índices actuales de los perfiles de Cantonera y Diagonal $[idx_c, idx_d]$.
- **Salida (Output Layer):** La capa final entrega un vector de valores-Q estimados, correspondientes a la utilidad esperada para cada una de las 4 acciones discretas posibles (Aumentar/Disminuir perfil de Cantonera o Diagonal). El agente selecciona la acción con el valor máximo ($\operatorname{argmax}_a Q(s, a)$) durante la fase de explotación.

A continuación, se describen los componentes específicos del algoritmo de entrenamiento seleccionados para este estudio:

- **Optimizador Adam:** Se utiliza el algoritmo de Estimación de Momentos Adaptativos (Adam) para la actualización de los pesos de la red. Este optimizador es ampliamente utilizado en aprendizaje profundo debido a su eficiencia computacional y su capacidad para ajustar tasas de aprendizaje individuales, facilitando la convergencia en problemas con gradientes ruidosos (Fu et al., 2024, Chen et al., 2019).
- **Función de Pérdida (MSELoss):** Para cuantificar el error en la predicción de los valores Q, se emplea el Error Cuadrático Medio (Mean Squared Error). Esta función penaliza las discrepancias grandes entre el valor predicho por la red y el valor objetivo (Target Q), guiando la optimización hacia una estimación más precisa de la recompensa futura (Sun et al., 2020).

Para garantizar la reproducibilidad de los resultados, en la Tabla 1 se especifican los hiperparámetros utilizados en el entrenamiento, implementados mediante la librería *PyTorch*:

Tabla 1: Configuración de la Red Neuronal y el Agente.

Parámetro	Valor / Configuración
Arquitectura de Red	MLP (Fully Connected)
Dimensiones de Capas	[Entrada: 2] → [Oculta: 64] → [Oculta: 64] → [Salida: 4]
Función de Activación	ReLU (en capas ocultas)
Optimizador	Adam (Adaptive Moment Estimation)
Función de Pérdida	MSELoss (Error Cuadrático Medio)
Tamaño del Lote (Batch Size)	64 muestras
Memoria de Repetición	2000 transiciones (Experience Replay Buffer)
Factor de Descuento (γ)	0.95
Pasos por Episodio	50 (Límite máximo para evitar bucles infinitos)

4. Ejemplo Ilustrativo

Para validar la metodología propuesta y calibrar los hiperparámetros del agente antes de abordar el problema tridimensional completo, se desarrolló un entorno de simulación controlado en 2D. El modelo representa una de las caras laterales de la estructura de soporte, simplificado a un marco plano en el eje X-Z, modelado e interactuado vía API con SAP2000. Esta fase permite verificar la hipótesis fundamental: que el agente DQN es capaz de converger al óptimo global en un espacio de búsqueda discreto, no convexo y sujeto a restricciones de rigidez.

Las dimensiones de la estructura son:

- H_t : 2.5 m
- h : 0.833 m

- B : 0.8 m

La Figura 4.1 muestra la geometría generada automáticamente por el script en SAP2000.

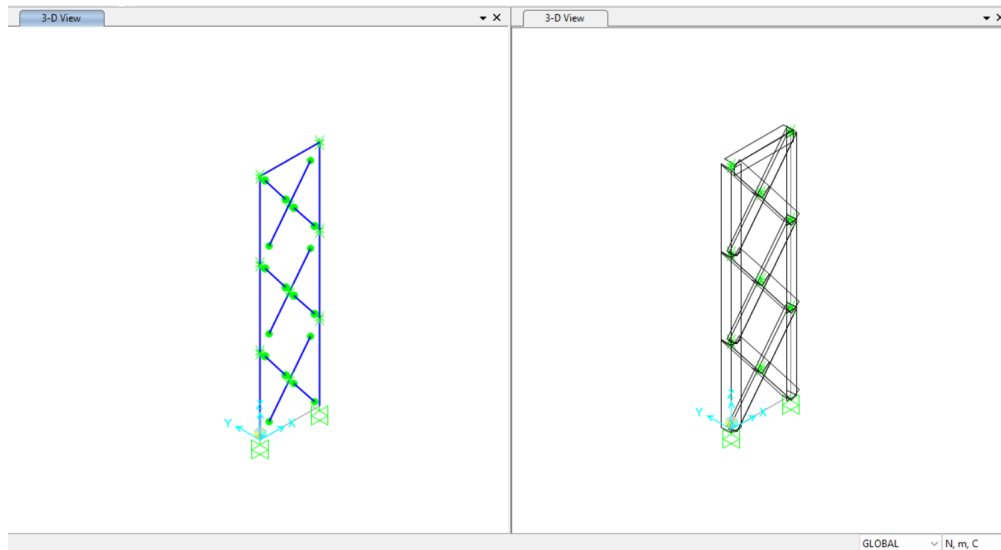


Figura 4.1: Modelo simplificado 2D en terminos de: vistas Offset (Panel izquierdo) y Extrude (Panel derecho) obtenidas de SAP2000

4.1. Definición del Espacio de Búsqueda y Materiales

A diferencia de los problemas de optimización continua, el diseño estructural utiliza perfiles comerciales estandarizados. Se definió un catálogo discreto C compuesto por 40 perfiles tipo Ángulo (L), variando desde un L40x4 hasta un L100x12. Los perfiles fueron ordenados estrictamente por su peso lineal (kg/m), de modo que el índice $i = 0$ corresponde al perfil más liviano y $i = 39$ al más pesado.

- **Material:** Acero ASTM A36 ($F_y = 250$ MPa).
- **Variables de Diseño:** El problema se reduce a dos variables enteras independientes:
 1. idx_c : Índice del perfil asignado a las Columnas.
 2. idx_d : Índice del perfil asignado a las Diagonales.

- **Tamaño del Espacio:** $40 \times 40 = 1,600$ combinaciones posibles.

La tabla completa de perfiles utilizados se encuentra detallada en el Anexo C.

4.2. Cargas y Criterios de Diseño

Para la evaluación del modelo plano en el eje X-Z, se aplicó el método de Diseño por Estados Límites Últimos (LRFD). En concordancia con las hipótesis de diseño adoptadas para esta investigación (ver Sección 2.1.3), se desestiman las cargas meteorológicas (CA y CA_s) bajo la premisa de que una estructura reticulada liviana capaz de resistir demandas sísmicas de alta magnitud ($A_0 = 0,5g$) satisface también los requerimientos de viento en las zonas de emplazamiento. En consecuencia, el agente verifica la resistencia de los elementos bajo la envolvente (ENV) que agrupa los siguientes estados límites, considerando la reversibilidad del sismo:

- **CU1 (Operación Normal)**
- **CU2 (Sismo Severo)**
- **CU4 (Sismo con Levantamiento)**

Esta formulación asegura que el agente explore soluciones robustas tanto para la compresión máxima (CU2) como para la tracción o levantamiento en las fundaciones (CU4).

4.3. Solución Exacta: Método de Fuerza Bruta

Para tener un mejor entendimiento del problema de optimización, se ejecutó una evaluación exhaustiva de todas las configuraciones estructurales posibles.

En la Figura 4.2 se presenta una caracterización completa del espacio de búsqueda para el modelo 2D. El gráfico de la izquierda muestra las curvas de nivel de la función objetivo (Peso), evidenciando que el peso aumenta de manera monótona hacia la esquina superior derecha, donde se concentran los perfiles más robustos. La línea roja gruesa representa la frontera crítica de diseño

definida por la restricción de frecuencia ($f = 30Hz$). Se observa claramente que esta restricción actúa como una barrera no lineal que divide el espacio en dos regiones: una zona no factible (a la izquierda de la línea roja, donde $f < 30Hz$) y una zona factible (a la derecha).

El gráfico de la derecha confirma la discretización de los diseños evaluados, donde las cruces rojas indican configuraciones que no cumplen con los criterios normativos, mientras que los círculos verdes representan diseños válidos. Del total de 1,600 combinaciones posibles, solo 532 resultaron factibles, lo que corresponde a un 33.2 % del espacio de búsqueda. Este porcentaje relativamente bajo subraya la dificultad que enfrentaría un método de búsqueda aleatoria pura. La solución óptima global, marcada con una estrella amarilla, se localiza exactamente en el límite de la frontera de factibilidad, lo cual es consistente con la teoría de optimización estructural.

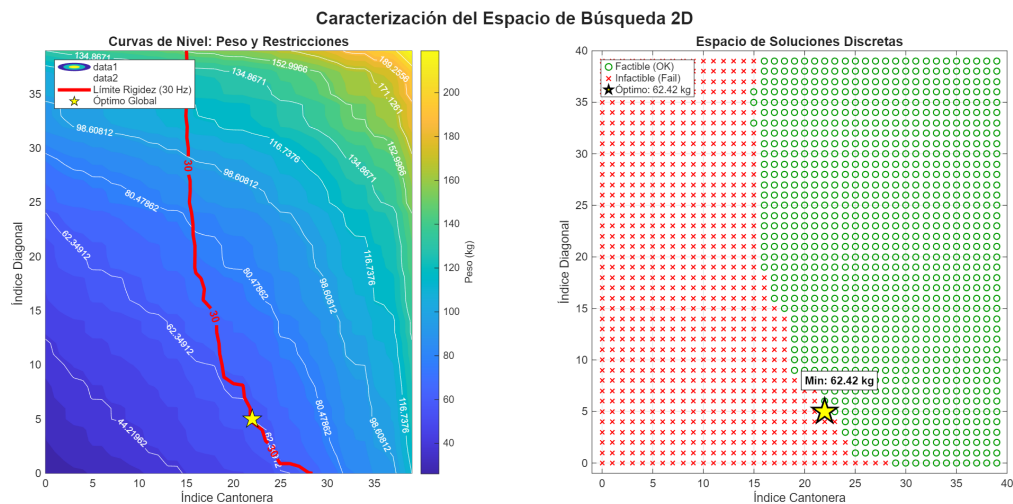


Figura 4.2: Izquierda: Mapa de calor y curvas de nivel del espacio de búsqueda de las soluciones. Derecha: Espacio de soluciones discretas de los diseños evaluados.

Los resultados numéricos precisos de este análisis exhaustivo se detallan en la Tabla 2.

Tabla 2: Solución Óptima Global obtenida por Fuerza Bruta.

Parámetro	Valor Óptimo	Descripción del Perfil
Variable Cantonera (idx_c)	22	L75x6
Variable Diagonal (idx_d)	5	L55x4
Peso Mínimo (W^*)	62.42 kg	Objetivo a alcanzar
Frecuencia Fundamental	30.12 Hz	Cumple (≥ 30 Hz)
Factor Utilización Máx.	0.054	Cumple ($\leq 1,0$)

De la tabla se desprende una conclusión fundamental para el diseño de este tipo de soportes: la solución óptima está gobernada casi exclusivamente por la rigidez. La frecuencia fundamental del diseño ganador es de 30.12 Hz, apenas un 0.4 % por encima del límite normativo, mientras que el Factor de Utilización máximo es extremadamente bajo (0.054), indicando que la estructura trabaja muy por debajo de su capacidad resistente. Esto confirma que el desafío principal para el agente de IA no será asegurar la resistencia de los materiales, sino navegar la restricción de rigidez para encontrar el diseño más liviano posible sobre los 30 Hz.

4.4. Desempeño del Algoritmo DQN

Para evaluar la capacidad de aprendizaje del agente bajo diferentes condiciones, se implementó el algoritmo DQN descrito en la Sección 2.3. La arquitectura interna de la red neuronal corresponde a la detallada previamente en la Tabla 1.

4.4.1. Definición de Escenarios Experimentales

Se diseñaron tres escenarios de entrenamiento distintos variando los hiperparámetros clave que controlan la exploración y la velocidad de aprendizaje. El objetivo es analizar cómo estas variables afectan la convergencia hacia el óptimo global. Los parámetros específicos se detallan en la Tabla 3.

Tabla 3: Definición de Escenarios de Entrenamiento.

Parámetro	Escenario A (Base)	Escenario B (Explorador)	Escenario C (Conservador)
Tasa de Aprendizaje (α)	0,001	0,005	0,0005
Decaimiento ϵ	0,95	0,99	0,90
Estrategia de Inicio	Aleatoria	Aleatoria	Fija (Perfil Máximo)

- **Escenario A (Base):** Representa una configuración estándar con una tasa de aprendizaje moderada y un decaimiento de exploración equilibrado.
- **Escenario B (Explorador):** Se caracteriza por una tasa de aprendizaje más alta y un decaimiento de ϵ muy lento (0,99), lo que fuerza al agente a explorar el espacio de soluciones durante más tiempo antes de explotar el conocimiento adquirido.
- **Escenario C (Conservador):** Simula un enfoque de diseño tradicional "seguro", iniciando la búsqueda desde el perfil más robusto del catálogo ($idx = 39$) e intentando reducir el peso progresivamente con una tasa de aprendizaje baja.

4.4.2. Entorno Computacional

Todos los experimentos numéricos, incluyendo las simulaciones de fuerza bruta y el entrenamiento de los agentes, se ejecutaron en un equipo portátil HP Victus. Las especificaciones de hardware relevantes para la reproducibilidad de los tiempos de cómputo reportados son:

- **Procesador:** Intel Core i5-12450H (12^a Generación).
- **Núcleos:** 8 núcleos, con una frecuencia turbo máxima de 2 GHz.
- **Memoria RAM:** 8 GB DDR4 a 3200 MT/s.
- **GPU:** NVIDIA GeForce RTX 2050.

- **Sistema Operativo:** Windows 11 Home (64 bits).

Es importante destacar que la comunicación con la API de SAP2000 se realiza de forma secuencial en un solo hilo de CPU, lo que representa el principal cuello de botella en el tiempo de ejecución total.

4.4.3. Resultados de una Ejecución Típica

La Tabla 4 muestra el mejor diseño encontrado por el agente en una sola ejecución de entrenamiento independiente. Los resultados indican que el agente logró identificar exactamente la combinación óptima de perfiles, demostrando su capacidad para navegar la no-linealidad de la restricción de frecuencia.

Tabla 4: Mejor Diseño encontrado por Agente DQN (Escenario Base).

Métrica	Resultado DQN	Diferencia con Óptimo
Perfiles Seleccionados	L75x5 / L55x4	Mismos perfiles
Peso Estructural	62.42 kg	0.00 %
Frecuencia	30.12 Hz	-
Episodios para Converger	18	-
Tiempo de Cómputo	≈ 12 min	15 % del tiempo de Fuerza Bruta

4.4.4. Interpretación de la Dinámica de Aprendizaje

La Figura 4.3 ilustra el proceso de convergencia típico de un agente exitoso (Escenario A).

- **Fase de Descenso Rápido (Episodios 1-5):** Se observa una caída abrupta en el peso estructural (línea roja), pasando de configuraciones iniciales aleatorias o pesadas a la vecindad del óptimo en muy pocos episodios. Esto demuestra que el gradiente de la función de recompensa guía eficazmente al agente hacia zonas de bajo peso.

- Fase de Refinamiento y Exploración (Episodios 5-45):** Una vez alcanzada la zona factible de bajo peso (≈ 64 kg), el agente oscila explorando soluciones vecinas (puntos verdes dispersos). Aunque la trayectoria del mejor peso (línea roja) se mantiene plana en el óptimo, el agente continúa explotando posibles diseños para asegurar que no existe una solución mejor, lo que se refleja en la alta variabilidad de la recompensa promedio (gráfico derecho) durante esta etapa.

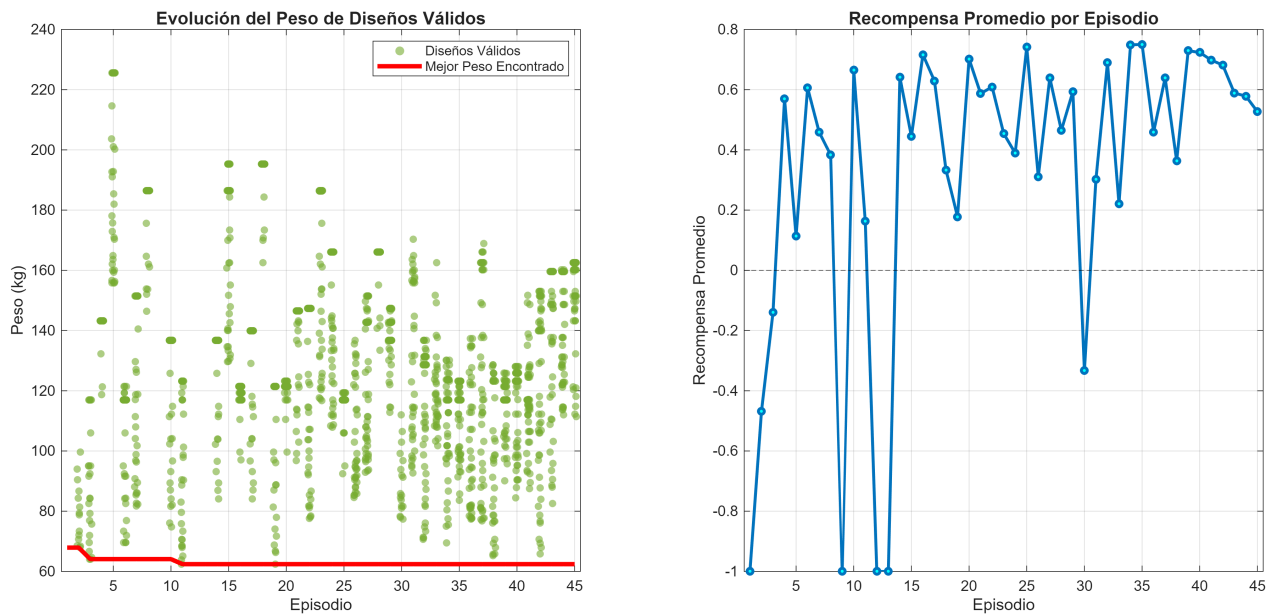


Figura 4.3: Dinámica de una iteración exitosa (Modelo 2D). Izquierda: Evolución del peso de diseños válidos y trayectoria del mejor encontrado. Derecha: Variación de la recompensa promedio.

4.5. Resultados Estadísticos y Dinámica de Aprendizaje

Para caracterizar la robustez y confiabilidad de los agentes, se desarrolla una evaluación estadística, los resultados obtenidos en 100 ejecuciones independientes de método propuesto. La Tabla 5 resume las métricas de desempeño para los tres escenarios propuestos, contrastados con el óptimo global teórico de 62,42 kg obtenido por fuerza bruta.

Tabla 5: Resumen Estadístico del Desempeño (100 Ejecuciones independientes - Modelo 2D).

Escenario	Tasa de Éxito	Peso Promedio	Desv. Est.	Gap Promedio	Promedio Episodios
A: Base	51.0 %	64.80 kg	3.76 kg	3.82 %	17.9
B: Explorador	52.0 %	64.49 kg	3.44 kg	3.32 %	16.4
C: Conservador	0.0 %	106.02 kg	25.08 kg	69.85 %	22.4

*Tasa de Éxito: % de corridas que convergen al óptimo global exacto (62.42 kg).

Los resultados diferencian dualidad en el comportamiento de los agentes dependiendo de su punto de partida y estrategia de exploración.

1. **Equivalencia entre Base y Explorador:** Los escenarios A y B presentan un desempeño estadísticamente similar (Tasas de éxito del 51 % y 52 % respectivamente). Esto sugiere que la topología del espacio de búsqueda 2D es lo suficientemente convexa en la región factible como para que una exploración moderada (Escenario A) sea suficiente. Ambos logran un *Gap Promedio* inferior al 4 %, indicando que cuando no encuentran el óptimo global, convergen a soluciones subóptimas de muy alta calidad.
2. **Fallo Estructural del Enfoque Conservador:** El Escenario C presenta una tasa de éxito nula (0 %). Al iniciar la búsqueda desde los perfiles más robustos ($idx = 39$), el agente queda atrapado en mínimos locales de peso alto (promedio 106 kg). La alta desviación estándar (25.08 kg) indica un comportamiento errático; el agente es incapaz de encontrar una ruta descendente consistente que reduzca el peso sin violar momentáneamente la restricción de frecuencia ($f \geq 30$ Hz), lo que resulta en penalizaciones severas que desincentivan la búsqueda de soluciones más livianas.
3. **Eficiencia Computacional:** Los agentes A y B convergen rápidamente, requiriendo en promedio menos de 18 episodios para estabilizarse (gracias al criterio de *Early Stopping*). Esto representa una reducción de órdenes de magnitud en el costo computacional comparado con la evaluación exhaustiva del espacio.

4.5.1. Análisis de la Velocidad de Convergencia

Además de los valores finales, es crítico evaluar la velocidad con la que los agentes aprenden políticas efectivas. La Figura 4.4 presenta la evolución de la mediana del peso estructural a lo largo de los 100 episodios de entrenamiento.

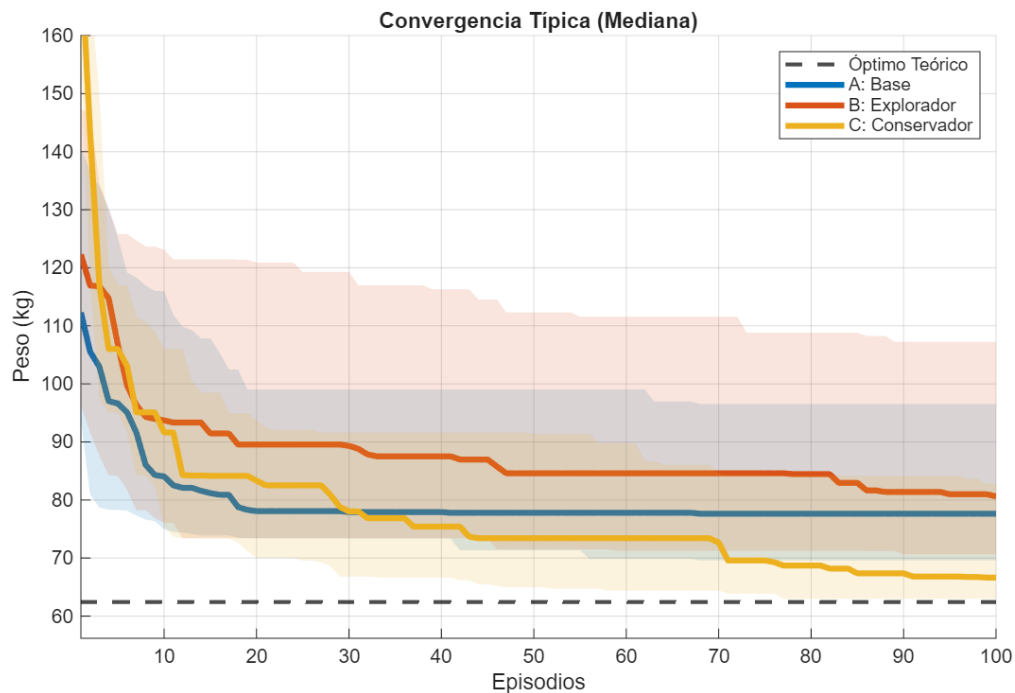


Figura 4.4: Curvas de Convergencia (Mediana de 100 ejecuciones independientes). Las áreas sombreadas representan el rango de variabilidad (desviación estándar) de cada escenario.

Del análisis gráfico se desprenden las siguientes observaciones:

1. **Eficiencia de Aprendizaje Temprano (Escenarios A y B):** Tanto el agente Base (azul) como el Explorador (rojo) logran reducir el peso estructural drásticamente en los primeros 10 a 15 episodios. La curva muestra una asíntota rápida, lo que indica que el algoritmo DQN identifica las características principales de la solución óptima con muy pocas interacciones con el entorno, validando la eficiencia del método para pre-diseño rápido.
2. **Estancamiento en Óptimos Locales:** Es relevante notar que la línea de la mediana no llega a tocar la línea discontinua del "Óptimo Teórico" (62.42 kg), sino que se estabiliza cerca de los

78 kg. Esto es consistente con la Tasa de Éxito del $\approx 50\%$ reportada en la Tabla 5. Significa que, en una "ejecución típica", el agente se aproxima al óptimo pero a menudo se detiene en un diseño muy eficiente pero que no es el óptimo global.

3. **Inercia del Agente Conservador (Escenario C):** La curva amarilla inicia en valores muy altos (> 160 kg) y desciende mucho más lentamente, estabilizándose recién cerca del episodio 70. Esto demuestra visualmente el costo computacional de iniciar con perfiles sobredimensionados: el agente pierde gran parte del tiempo de entrenamiento simplemente tratando de "adelgazar" la estructura de manera segura, en lugar de explorar configuraciones eficientes desde el inicio.

5. Ejemplo Aplicación

Tras validar la capacidad del agente para navegar espacios discretos en el modelo plano, se procede a la aplicación final sobre la torre reticulada completa. Esta etapa introduce una complejidad física significativamente mayor debido a la interacción de fuerzas bidireccionales, la torsión accidental y los modos de vibración acoplados, manteniendo el mismo tamaño del espacio combinatorio (1,600 estados).

La geometría corresponde a una torre de base cuadrada ($0,80 \times 0,80$ m) extendida del modelo anterior (ver Figura 5.1). Las propiedades de los materiales y el catálogo de perfiles (Anexo C) se mantienen idénticos al caso 2D.

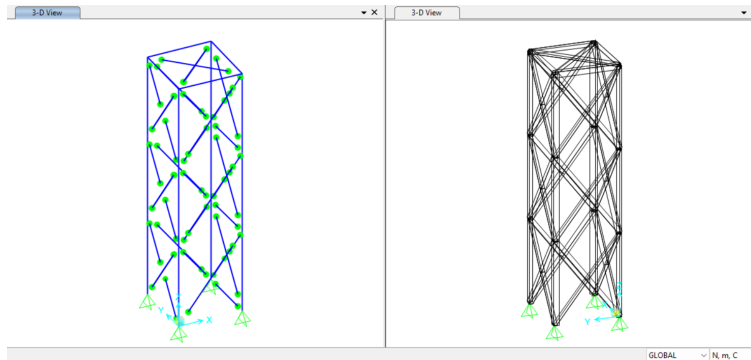


Figura 5.1: Modelo 3D en SAP2000 considerando efectos espaciales.

A diferencia del modelo plano, las solicitaciones se rigen por la envolvente completa de la norma sísmica, considerando la acción simultánea en tres ejes ortogonales (E_x, E_y, E_z) y la excentricidad accidental de la masa. Las combinaciones de carga implementadas en la API de SAP2000 siguen la regla de combinación direccional ($100\%X + 30\%Y$), lo que impone demandas de compresión biaxial en las columnas esquineras que no existían en el modelo 2D.

5.1. Análisis del Espacio de Búsqueda 3D

Para comprender la naturaleza del problema antes de aplicar el aprendizaje por refuerzo, se realizó un barrido exhaustivo (fuerza bruta) de todas las configuraciones estructurales posibles. Los resultados de este análisis se presentan en la Figura 5.2, donde se visualizan las curvas de nivel de peso y la distribución de soluciones discretas.

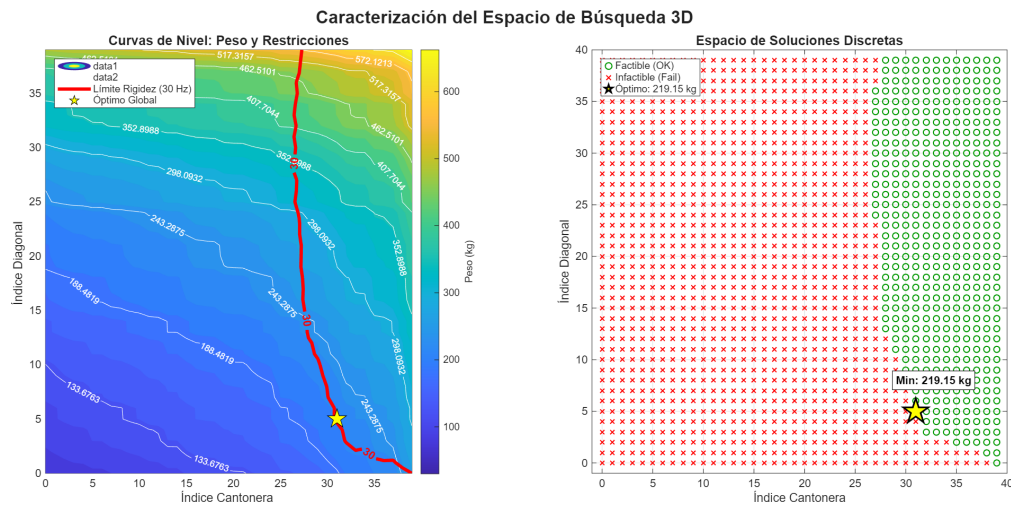


Figura 5.2: Caracterización del Espacio de Búsqueda 3D. Izquierda: Mapa de calor del peso con la frontera de restricción de frecuencia (30 Hz). Derecha: Clasificación binaria de diseños factibles (verde) e infactibles (rojo).

Los resultados numéricos exactos se pueden ver en la Tabla 6.

Tabla 6: Solución Óptima Global obtenida por Fuerza Bruta.

Parámetro	Valor Óptimo	Descripción del Perfil
Variable Cantonera (idx_c)	31	L90x8
Variable Diagonal (idx_d)	5	L55x4
Peso Mínimo (W^*)	219.15 kg	Objetivo a alcanzar
Frecuencia Fundamental	30.15 Hz	Cumple (≥ 30 Hz)
Factor Utilización Máx.	0.126	Cumple ($\leq 1,0$)

Del análisis gráfico y numérico se desprenden dos observaciones fundamentales que diferencian este caso del modelo 2D:

1. **Contracción de la Región Factible:** Al comparar con el modelo plano, se observa una reducción notable en el número de diseños válidos. Mientras que en 2D el 33,2% de las combinaciones cumplían con los requisitos normativos, en el modelo 3D este porcentaje desciende al 27,5%. Esta reducción se atribuye a la aparición de modos de vibración torsionales

de baja frecuencia, los cuales no existían en el plano y que obligan a descartar combinaciones de perfiles ligeros que anteriormente eran aceptables.

2. **Incremento No Lineal del Peso:** Para una misma configuración de perfiles (mismos índices de catálogo C), el peso total de la estructura 3D no es simplemente el doble o cuádruple del modelo 2D, sino que aumenta de manera más agresiva debido a la necesidad de elementos adicionales de arriostramiento y a la mayor longitud de pandeo efectiva en las cantoneras. El óptimo global se desplaza hacia la derecha en el gráfico, pasando de 62,42 kg a 219,15 kg. Este aumento refleja la necesidad de resistir solicitaciones de flexión biaxial ($M_{ux} + M_{uy}$) en las cantoneras, una demanda estructural que penaliza severamente el uso de perfiles esbeltos.

5.2. Resultados Estadísticos y Desempeño del Agente 3D

Para caracterizar el desempeño del algoritmo, se estableció primero el resultado mediante fuerza bruta, identificando un Óptimo Global de 219.15 kg. Posteriormente, se realizaron 100 ejecuciones independientes para cada escenario. Los resultados se resumen en la Tabla 7.

Tabla 7: Resumen Estadístico del Desempeño (100 Corridas - Modelo 3D).

Escenario	Tasa de Éxito	Peso Promedio	Desv. Est.	Gap Promedio	Promedio Episodios
A: Base	72.0 %	221.82 kg	5.42 kg	1.22 %	39.0
B: Explorador	80.0 %	222.34 kg	9.24 kg	1.46 %	38.9
C: Conservador	8.0 %	280.26 kg	66.34 kg	27.89 %	44.9

*Tasa de Éxito considerando una tolerancia del 1.0 % respecto al óptimo (219.15 kg).

Los resultados indican que:

1. **Superioridad de la Exploración (Escenario B):** A diferencia del modelo 2D, donde los resultados fueron mixtos, en el entorno 3D el Escenario Explorador (B) demostró una superioridad clara, alcanzando una **Tasa de Éxito del 80 %**. Esto valida la hipótesis de que, ante

espacios de búsqueda con múltiples restricciones no lineales (torsión + pandeo) fomentando la exploración inicial, siendo esto crítico para evitar quedar atrapado en configuraciones subóptimas que satisfacen la resistencia pero penalizan el peso.

2. **Robustez del Agente Base (Escenario A):** El agente con parámetros estándar logró un desempeño notable (72 % de éxito) y la menor desviación estándar (5,42 kg). Esto indica que es una estrategia conservadora pero segura: falla menos catastróficamente que el explorador, aunque encuentra el óptimo global con menor frecuencia.
3. **Inviabilidad del Enfoque Conservador (Escenario C):** Se confirma que iniciar la optimización desde perfiles pesados es ineficaz. Con un éxito del 8 % y un Gap promedio cercano al 28 %, este agente demuestra incapacidad para navegar la barrera de frecuencia "hacia abajo", quedando estancado en diseños sobredimensionados.

6. Discusión Comparativa: 2D vs 3D

La transición del modelo plano al espacial permite extraer conclusiones fundamentales sobre la escalabilidad y comportamiento del algoritmo DQN en ingeniería estructural.

6.1. Contracción del Espacio de Soluciones

Al comparar la topología de ambos problemas, se observa una reducción de la región factible. Mientras que en 2D el 33.2 % de las combinaciones eran válidas, en 3D este número desciende al 27.5 %. Cualitativamente, esto se explica por la aparición de modos torsionales: combinaciones de perfiles que en 2D proveían suficiente rigidez lateral, en 3D resultan insuficientes para controlar la rotación de la estructura, siendo descartadas por la restricción de frecuencia ($f \geq 30$ Hz).

6.2. Evolución de la Dificultad de Convergencia

Se observa un cambio en la jerarquía de las estrategias de aprendizaje. En el modelo 2D (espacio convexo y suave), la exploración agresiva no aportaba ventajas significativas sobre la estrategia base. Sin embargo, en el modelo 3D (espacio rugoso con óptimos locales), la capacidad de exploración se volvió determinante, elevando la tasa de éxito del Escenario B por encima del A. Esto sugiere que, a medida que aumenta la complejidad del problema estructural, los algoritmos de IA deben configurarse con políticas de exploración más agresivas para garantizar resultados óptimos.

7. Conclusiones

Esta memoria de título explora la aplicación de técnicas de Aprendizaje por Refuerzo Profundo (DRL) al diseño óptimo de estructuras desoporte sujetas a la estricta normativa sísmica chilena. A partir de los resultados obtenidos, se concluye lo siguiente:

1. **Validación y Robustez del Algoritmo:** La metodología propuesta demostró ser eficaz para encontrar el óptimo global en espacios de búsqueda combinatorios. En el modelo 2D, el agente logró una tasa de convergencia del 52 %, mientras que en el modelo 3D, a pesar del aumento significativo en la complejidad física, el agente mantuvo una tasa de éxito del 80 % (Escenario B). Esto valida la hipótesis de que un agente de IA puede navegar espacios no convexos sin necesidad de gradientes.
2. **Superioridad de la Estrategia de Exploración:** El análisis estadístico confirmó que la estrategia de exploración es determinante. El *Escenario B (Explorador)* no solo igualó la tasa de convergencia del modelo base, sino que logró soluciones de mayor calidad, reduciendo el error relativo (Gap) al 1.22 % respecto al óptimo teórico. Esto demuestra que, en problemas estructurales discretos, fomentando la exploración inicial para evitar mínimos locales subóptimos.

3. **Eficiencia frente al Diseño Tradicional:** El algoritmo identificó una solución de **219.15 kg**, optimizando la distribución de material de manera no intuitiva. El agente “aprendió” que la estructura estaba controlada por rigidez ($f \geq 30$ Hz) y no por resistencia, asignando perfiles más robustos en las columnas y más livianos en las diagonales para cumplir la norma con el mínimo peso.

4. **Escalabilidad Potencial:** Si bien el costo computacional del agente DQN se mantiene lineal frente al crecimiento exponencial de la fuerza bruta, su mayor potencial radica en la arquitectura del vector de estado (s_t). Es factible escalar este vector para incluir no solo las secciones transversales, sino también parámetros topológicos y geométricos de la estructura. Esto permitiría llevar la metodología a una fase productiva, donde el agente no necesite re-entrenarse para cada nuevo proyecto, sino que actúe como un diseñador generalista capaz de inferir soluciones óptimas para diversas tipologías y geometrías estructurales basándose en una política de diseño aprendida previamente.

A. Anexos: Algoritmo de Optimización

En este apéndice se presentan los fragmentos esenciales del código desarrollado para la implementación de la metodología en el núcleo de optimización e interacción con SAP2000 (escrito en Python). El script principal gestiona la conexión con la API de SAP2000, la definición del Agente DQN y el bucle de entrenamiento.

A.1. Entorno Virtual y su Interacción con SAP2000

La clase `VirtualSapEnv` actúa como intermediario. Traduce las acciones del agente (cambiar perfiles) en comandos OAPI y extrae los resultados del análisis (Peso, Frecuencia, FU) para calcular la recompensa.

```
1 class VirtualSapEnv:
2     def step(self, action):
3         # 1. Decodificar la accion
4         # (0: Aumentar Columna, 1: Disminuir Columna, etc.)
5         s = list(self.state)
6         if action == 0: s[0] = min(self.n_profiles-1, s[0]+1)
7         elif action == 1: s[0] = max(0, s[0]-1)
8         # ... (logica para diagonales) ...
9
10        self.state = s
11
12        # 2. Consultar "Simulador" (Base de datos o SAP2000 directo)
13        # d contiene: Peso (w), Frecuencia (f), Ratio (r)
14        d = self.lookup_table.get((s[0], s[1]))
15
16        # 3. Calcular Recompensa Normalizada
17        if not d['ok']:
18            # Penalizacion por diseno infactible
19            reward = -1.0
```

```

20     else:
21         # Recompensa por diseno factible
22         # Puntuacion de Peso (Lineal 0 a 1)
23         denom = (self.w_max - self.w_min)
24         s_w = (self.w_max - d['w']) / denom
25
26
27         # Puntuacion de Frecuencia (Inversa Simple)
28         # Incentiva acercarse al limite de 30Hz sin excederse
29         s_f = 30.0 / d['f']
30
31         # Combinacion Ponderada
32         reward = (0.7 * s_w) + (0.3 * s_f)
33
34     return np.array(self.state), reward, False, d

```

Listing 1: Lógica de interacción con SAP2000 y cálculo de Recompensa

A.2. Agente DQN

La clase DQNAgent implementa la red neuronal y la política de decisión ϵ -greedy.

```

1 class DQNAgent:
2     def __init__(self, state_size, action_size, params):
3         # Red Neuronal Profunda (Estimador de Q)
4         self.model = nn.Sequential(
5             nn.Linear(state_dim, 64), nn.ReLU(),
6             nn.Linear(64, 64), nn.ReLU(),
7             nn.Linear(64, action_dim)
8         )
9         # Optimizador Adam
10        self.opt = optim.Adam(self.model.parameters(), lr=params['LR'])
11        # Memoria de Repeticion

```

```

12     self.memory = deque(maxlen=2000)
13
14     def act(self, state):
15         # Politica Epsilon-Greedy
16         if random.random() <= self.epsilon:
17             return random.randrange(self.action_size) # Exploracion
18         else:
19             with torch.no_grad():
20                 q_values = self.model(torch.FloatTensor(state))
21                 return torch.argmax(q_values).item() # Explotacion
22
23     def replay(self):
24         # Aprendizaje por lotes (Experience Replay)
25         batch = random.sample(self.memory, self.batch_size)
26         # ... (Calculo de perdida y Backpropagation) ...
27         loss.backward()
28         self.opt.step()

```

Listing 2: Estructura del Agente DQN

B. Diagrama de Flujo del Algoritmo

A continuación se presenta el diagrama de flujo que describe el proceso iterativo del agente DQN dentro de un episodio de entrenamiento. Este esquema detalla la interacción cíclica entre la toma de decisiones, la simulación física y la actualización de la red neuronal.

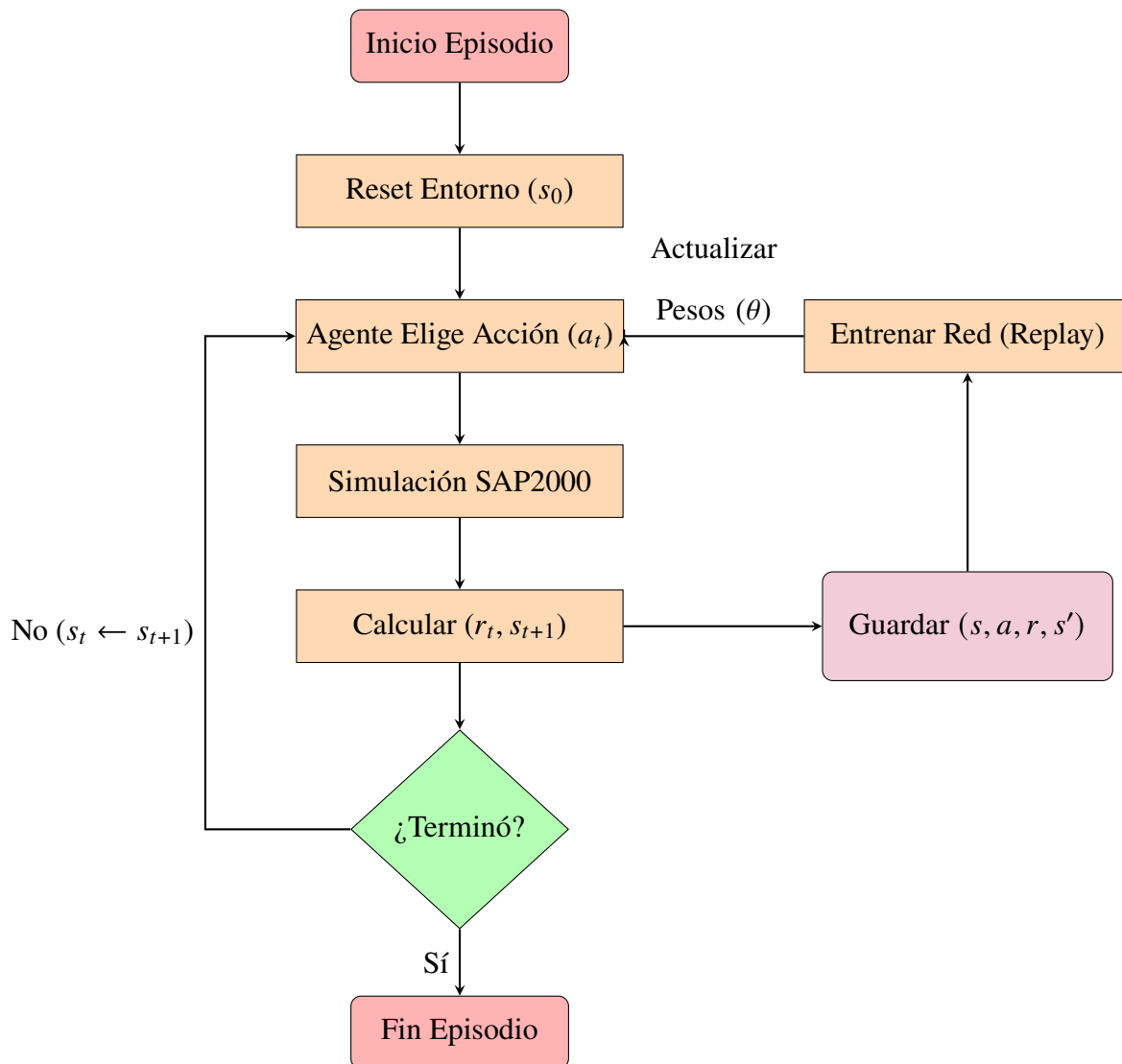


Figura B.1: Diagrama de Flujo del Bucle de Entrenamiento por Episodio.

Descripción del Proceso

1. **Inicialización:** Se configura la estructura con perfiles iniciales (aleatorios o fijos).

2. **Acción:** La red neuronal evalúa el estado actual (s_t) y selecciona una acción (a_t) (cambiar perfil) usando una política ϵ -greedy. (Ver Sección 3.2.2).
3. **Simulación:** Se actualiza el modelo en SAP2000 y se ejecuta el análisis de elementos finitos.
4. **Recompensa:** Se calcula r_t basada en el peso y cumplimiento normativo.(Ver Sección 3.3).
5. **Memoria y Aprendizaje:** La experiencia se guarda en el búfer. Paralelamente, se extrae un lote aleatorio de experiencias pasadas para entrenar la red neuronal (Backpropagation), refinando la toma de decisiones para pasos futuros.
6. **Bucle:** Si el episodio no ha terminado (por límite de pasos o convergencia), el estado se actualiza y el ciclo se repite.

C. Catálogo de Perfiles Estructurales

La Tabla 8 detalla el espacio de búsqueda discreto utilizado tanto para el modelo 2D como para el 3D. Los perfiles corresponden a ángulos estructurales de alas iguales (L) según norma, ordenados por peso lineal ascendente para facilitar la navegación del algoritmo de optimización.

Tabla 8: Catálogo de Perfiles (Ordenado por Peso)

Índice	Nombre Perfil	Área (cm^2)	Peso (kg/m)
0	L45X3	2.62	2.06
1	L40X4	3.05	2.40
2	L45X4	3.45	2.71
3	L40X5	3.78	2.97
4	L50X4	3.85	3.02
5	L55X4	4.25	3.34
6	L45X5	4.28	3.36
7	L60X4	4.65	3.65
8	L50X5	4.78	3.75
9	L45X6	5.08	3.99
10	L55X5	5.28	4.14
11	L50X6	5.68	4.46
12	L60X5	5.78	4.54
13	L55X6	6.28	4.93
14	L65X5	6.28	4.93
15	L60X6	6.88	5.40
16	L75X5	7.28	5.72
17	L50X8	7.43	5.83
18	L65X6	7.48	5.87

Índice	Nombre Perfil	Área (cm²)	Peso (kg/m)
19	L70X6	8.08	6.34
20	L55X8	8.13	6.38
21	L50X9	8.28	6.50
22	L75X6	8.68	6.82
23	L60X8	9.03	7.09
24	L80X6	9.28	7.29
25	L90X6	10.50	8.24
26	L70X8	10.60	8.32
27	L60X10	11.10	8.71
28	L100X6	11.70	9.18
29	L80X8	12.20	9.58
30	L70X10	13.10	10.28
31	L90X8	13.80	10.83
32	L80X10	15.10	11.85
33	L100X8	15.40	12.09
34	L90X9	15.50	12.17
35	L90X10	17.10	13.42
36	L100X9	17.30	13.58
37	L80X12	17.90	14.05
38	L90X12	20.30	15.93
39	L100X12	22.70	17.82

Referencias Bibliográficas

- American Society of Civil Engineers. (2015). *Design of Latticed Steel Transmission Structures* (inf. téc. N.º ASCE 10-15). ASCE.
- Belegundu, A. D., & Chandrupatla, T. R. (2011). *Optimization concepts and applications in engineering*. Cambridge University Press. <https://doi.org/10.1017/cbo9780511975905>
- Chen, S.-A., Tangkaratt, V., Lin, H.-T., & Sugiyama, M. (2019). Active deep Q-learning with demonstration. *Machine Learning*, 109(9-10), 1699-1725. <https://doi.org/10.1007/s10994-019-05849-4>
- Comisión Nacional de Energía. (2019). *Anexo Técnico: Exigencias Mínimas para el Diseño de Instalaciones de Transmisión* (inf. téc.) (Norma Técnica de Seguridad y Calidad de Servicio (NTSyCS)). Comisión Nacional de Energía. Santiago, Chile.
- Comité Chileno de CIGRE. (2020). *Recomendación de Requisitos Sísmicos para Instalaciones Eléctricas de Alta Tensión* (inf. téc.). Comité Chileno de CIGRE. Santiago, Chile.
- Dworschak, F., Dietze, S., Wittmann, M., Schleich, B., & Wartzack, S. (2022). Reinforcement Learning for Engineering Design Automation. *Advanced Engineering Informatics*, 52, 101612. <https://doi.org/10.1016/j.aei.2022.101612>
- Fu, B., Gao, Y., & Wang, W. (2024). A physics-informed deep reinforcement learning framework for autonomous steel frame structure design. *Computer-Aided Civil and Infrastructure Engineering*. <https://doi.org/10.1111/mice.13276>
- Hansen, S. (2016). Using Deep Q-Learning to Control Optimization Hyperparameters. *arXiv preprint arXiv:1602.04062*. <http://arxiv.org/abs/1602.04062>
- Hertel, L., Baldi, P., & Gillen, D. L. (2020). Quantity vs. Quality: On Hyperparameter Optimization for Deep Reinforcement Learning. *arXiv preprint arXiv:2007.14604*. <http://arxiv.org/abs/2007.14604>
- Honda, S., Imura, Y., Sasaki, K., & Takeda, R. (2022). Adaptive Vibration Control of Smart Structure Using Deep Reinforcement Learning. *EPI International Journal of Engineering*, 5(2), 92-97. <https://doi.org/10.25042/epi-ije.082022.03>
- Hu, Z., Wan, K., Gao, X., & Zhai, Y. (2019). A Dynamic Adjusting Reward Function Method for Deep Reinforcement Learning with Adjustable Parameters. *Mathematical Problems in Engineering*, 2019. <https://doi.org/10.1155/2019/7619483>
- Instituto Nacional de Normalización. (2023). *Diseño sísmico de estructuras e instalaciones industriales* (inf. téc. N.º NCh2369.Of2023). INN. Santiago, Chile.
- Millan, C., & Filho, J. (2019). Modified simulated annealing algorithm for optimal design of steel structures. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 35(1). <https://doi.org/10.23967/j.rimni.2019.03.003>
- Sun, H., Burton, H. V., & Huang, H. (2020). Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering*, 33, 101816. <https://doi.org/10.1016/j.jobe.2020.101816>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2.ª ed.). MIT Press.
- Transec S.A. (2013). *Especificaciones de Diseño Sísmico de Instalaciones Eléctricas de Alta Tensión* (inf. téc. N.º ETG-A.0.20, Modificación 8). Transec S.A.

Wei, S., Bao, Y., & Li, H. (2020). Optimal policy for structure maintenance: A deep reinforcement learning framework. *Structural Safety*, 83, 101906. <https://doi.org/10.1016/j.strusafe.2019.101906>