Universidad del Desarrollo
Facultad de Ingeniería

OPTIMIZACIÓN DE RUTAS BAJO INCERTIDUMBRE MEDIANTE APRENDIZAJE POR REFUERZO Y METAHEURÍSTICAS SENSIBLES AL RIESGO

Un enfoque híbrido ALNS–PPO para el ruteo de vehículos con ventanas de tiempo y evaluación mediante CVaR

POR: GABRIEL ALEJANDRO ÁLVAREZ MARTÍNEZ-CONDE

Proyecto de grado presentado a la Facultad de Ingeniería de la Universidad del Desarrollo para optar al grado académico de Magíster en Data Science

PROFESOR GUÍA:

Sr. MAURICIO RENÉ HERRERA MARÍN

Enero 2026

SANTIAGO

A mi madre que hoy vuelve a estar presente en mis plantas, flores y en mi pensamiento.
Un día a la vez.

AGRADECIMIENTO

Agradezco a Inés por su paciencia, apoyo mutuo y acompañamiento en este proceso, a Francisco por las largas conversaciones que ayudan a aclarar la mente, a Leonardo por su cariño incondicional, a Rodrigo por siempre tener una palabra da apoyo en alegrías y tristezas, a Claudia porque la distancia no es obstáculo para estar presente, a Rafael que me motiva a ser mejor persona, a mi familia que sin ellos nada de esto sería posible y a los amigos y próximamente colegas que hice durante el magíster que hicieron mucho más llevadero este tiempo.

TABLA DE CONTENIDO

Resumen

Este trabajo aborda la resolución de un problema de ruteo de vehículos con ventanas de tiempo bajo incertidumbre, incorporando explícitamente métricas de riesgo mediante simulación Monte Carlo. El objetivo principal es comparar distintos enfoques de optimización, evaluando no solo la calidad promedio de las soluciones, sino también su comportamiento en escenarios adversos, utilizando el Conditional Value at Risk al 95% (CVaR95) como métrica central de desempeño.

Se analizan tres métodos: un enfoque determinista basado en OR-Tools, un esquema metaheurístico de Adaptive Large Neighborhood Search (ALNS) y una extensión híbrida que integra ALNS con aprendizaje por refuerzo profundo mediante Proximal Policy Optimization (PPO). Las soluciones obtenidas se evalúan bajo múltiples escenarios de severidad creciente, considerando métricas de tardanza promedio, CVaR95, nivel de servicio y esfuerzo computacional, medido a través del tiempo de cómputo y la cantidad de ticks ejecutados.

Los resultados muestran que el enfoque determinista presenta limitaciones estructurales bajo incertidumbre, con altos niveles de riesgo extremo y bajo cumplimiento de ventanas de tiempo, a pesar de su bajo costo computacional. El ALNS clásico logra mejoras sustanciales en calidad y robustez, pero a costa de un esfuerzo computacional elevado. En contraste, el método PPO-ALNS alcanza sistemáticamente los menores valores de CVaR95 y mayores niveles de servicio, especialmente en escenarios severos, utilizando una fracción del esfuerzo computacional requerido por ALNS. El análisis del

comportamiento de operadores evidencia que el agente de aprendizaje por refuerzo aprende políticas especializadas y adaptativas, concentrando el uso de operadores de alto impacto y evitando exploraciones poco eficientes.

En conjunto, los resultados demuestran que la integración de aprendizaje por refuerzo en un marco ALNS permite mejorar simultáneamente la robustez de las soluciones y la eficiencia computacional, posicionando al enfoque PPO-ALNS como una alternativa particularmente adecuada para aplicaciones operacionales de ruteo bajo incertidumbre.

1. Introducción

En la logística moderna, la eficiencia en la planificación y ejecución de entregas constituye un factor crítico para la competitividad de las empresas. La creciente complejidad de las cadenas de suministro, junto con la necesidad de cumplir restricciones operacionales cada vez más estrictas, como ventanas de tiempo, capacidades de carga, tiempos de servicio y disponibilidad limitada de flota, ha convertido al problema de ruteo de vehículos en uno de los desafíos más relevantes dentro de la investigación operativa y la optimización combinatoria.

Uno de los problemas más estudiados en este contexto es el Problema de Ruteo de Vehículos con Ventanas de Tiempo y Capacidad (CVRPTW), el cual busca determinar un conjunto de rutas que minimicen el costo total de operación, asegurando el cumplimiento de restricciones de capacidad y de servicio a los clientes dentro de intervalos temporales predefinidos. Tradicionalmente, este tipo de problemas se ha abordado mediante enfoques deterministas, utilizando solvers exactos o cuasi-exactos como Gurobi o OR-Tools, los cuales han demostrado un desempeño competitivo en instancias bien definidas y de tamaño moderado.

Sin embargo, en escenarios reales, la planificación logística se ve afectada por múltiples fuentes de incertidumbre, tales como variaciones en los tiempos de viaje, congestión vial, condiciones climáticas o fluctuaciones en la demanda. Estas fuentes de incertidumbre pueden provocar que soluciones óptimas bajo supuestos deterministas presenten un desempeño deficiente en la práctica, manifestándose en atrasos,

incumplimientos de ventanas de tiempo o un uso ineficiente de la flota. En este contexto, los métodos deterministas pierden parte de su efectividad al no considerar explícitamente el riesgo asociado a escenarios adversos.

Como alternativa, se han propuesto enfoques más flexibles basados en metaheurísticas, entre las cuales destaca la Búsqueda Adaptativa de Grandes Vecindarios (ALNS). Este enfoque permite explorar de manera eficiente el espacio de soluciones mediante la destrucción y reparación parcial de rutas, adaptando dinámicamente la selección de operadores en función de su desempeño histórico. No obstante, la selección de operadores en ALNS suele depender de reglas heurísticas diseñadas manualmente, lo que puede limitar su capacidad de adaptación a distintos escenarios de incertidumbre.

En este trabajo se propone un enfoque híbrido que combina ALNS con Aprendizaje por Refuerzo (RL), donde un agente de RL es entrenado para seleccionar de manera inteligente los operadores de destrucción y reparación dentro del proceso ALNS. Adicionalmente, se incorpora un criterio de evaluación sensible al riesgo, basado en la métrica de Conditional Value at Risk (CVaR), con el objetivo de obtener soluciones más robustas frente a la variabilidad de los tiempos de viaje. De esta forma, se busca no solo minimizar el costo esperado, sino también mejorar la estabilidad y confiabilidad del desempeño logístico bajo incertidumbre.

2. Trabajo Relacionado

La integración de técnicas de aprendizaje por refuerzo y métodos heurísticos para resolver problemas de ruteo ha sido objeto de creciente interés en la literatura reciente, especialmente en contextos donde la incertidumbre juega un rol relevante.

En (Lin, Ghaddar, & Nathwani, 2022) se propone la resolución de un Electric Vehicle Routing Problem with Time Windows (EVRPTW) mediante aprendizaje por refuerzo profundo, donde se entrenan políticas de ruteo capaces de construir soluciones de manera secuencial. Este enfoque demuestra que los métodos basados en DRL pueden aprender estrategias competitivas sin necesidad de reglas heurísticas explícitas, aunque su escalabilidad y garantía de factibilidad siguen siendo desafíos relevantes.

En (Cao, Zhu, Lu, & Zhang, 2025) se aborda un problema de optimización de rutas y programación de barcos bajo incertidumbre en la demanda de carga, incorporando un criterio de riesgo basado en CVaR. En este trabajo se comparan métodos exactos (Gurobi) con un enfoque heurístico basado en ALNS, integrando además un modelo de aprendizaje por refuerzo para la predicción de la demanda, la cual actúa como entrada para los algoritmos de optimización. Los resultados muestran que los métodos heurísticos presentan una mayor flexibilidad frente a escenarios inciertos, especialmente cuando el problema escala en tamaño.

El trabajo presentado en (Zhang, 2024) introduce un marco de aprendizaje por refuerzo con restricciones, orientado a garantizar seguridad probabilística en tareas de control

bajo incertidumbre, utilizando métricas como CVaR. Si bien el enfoque no se centra específicamente en problemas de ruteo, proporciona fundamentos teóricos relevantes para la incorporación de criterios sensibles al riesgo dentro de la función de recompensa de agentes de RL.

En (Zong, Xia, Meng, & Li, 2024) se propone un enfoque end-to-end basado en RL para resolver un VRPTW, integrando directamente las restricciones y penalizaciones por tardanza dentro de la función de recompensa. Este trabajo destaca la capacidad de RL para manejar restricciones complejas en tiempos de inferencia acotados, aunque requiere un entrenamiento intensivo y presenta dificultades para garantizar factibilidad estricta en todas las instancias.

Finalmente, en (Aravena Cabrera, 2023) se estudia el EVRP considerando incertidumbre en la velocidad de los vehículos, formulando el problema como uno estocástico no lineal y resolviéndolo mediante algoritmos genéticos. La estrategia propuesta mejora la robustez de las soluciones respecto a enfoques deterministas, manejando el cumplimiento de restricciones de manera probabilística y evidenciando la importancia de incorporar explícitamente la incertidumbre en la modelación del problema.

En conjunto, estos trabajos evidencian que, si bien los enfoques basados en RL y metaheurísticas presentan ventajas en escenarios inciertos, aún existe un espacio relevante para explorar soluciones híbridas que combinen la factibilidad garantizada de métodos heurísticos con la adaptabilidad y capacidad de aprendizaje de RL, especialmente bajo criterios sensibles al riesgo.

3. Hipótesis y Objetivos

3.1. Hipótesis de trabajo

Una política de aprendizaje por refuerzo entrenada bajo un criterio sensible al riesgo, utilizada para la selección adaptativa de operadores en un esquema ALNS, permite reducir el riesgo operacional medido mediante CVaR95 de la tardanza y mejorar la estabilidad del desempeño bajo incertidumbre, en comparación con enfoques deterministas y metaheurísticos clásicos, sin incurrir en aumentos significativos de costo total, número de vehículos utilizados ni tiempo de cómputo.

3.2. Objetivo general

Cuantificar el trade-off entre costo y robustez al resolver un Problema de Ruteo de Vehículos con Ventanas de Tiempo y Capacidad (CVRPTW) bajo incertidumbre, mediante simulación Monte Carlo y métricas de riesgo, comparando el desempeño de OR-Tools, ALNS y un enfoque híbrido de aprendizaje por refuerzo para la selección de operadores integrado en ALNS para asegurar factibilidad.

3.3. Objetivos específicos

- Implementar un modelo base de CVRPTW determinista utilizando OR-Tools como referencia de resolución exacta bajo tiempo acotado.
- Desarrollar un solver ALNS capaz de manejar restricciones de capacidad, ventanas de tiempo, tiempos de servicio y reglas de factibilidad operacional.

- Diseñar un entorno de aprendizaje por refuerzo para la selección adaptativa de operadores de ALNS, incorporando información del estado de la solución y del progreso de la búsqueda.
- Incorporar un esquema de evaluación robusta mediante simulación Monte Carlo para estimar métricas de riesgo, en particular CVaR95 de la tardanza.
- Analizar y comparar los enfoques en términos de calidad de solución, riesgo, estabilidad, tiempo de cómputo y esfuerzo computacional medido en ticks, considerando distintos niveles de severidad de los escenarios.

4. Datos y Metodología

4.1. Descripción y preparación de los datos

Datos originales

El conjunto de datos base corresponde al dataset “Delivering Data: Atenas”, el cual contiene órdenes de despacho diarias, organizadas por día de operación. Cada día contiene un conjunto de clientes a atender desde un único depósito, junto con la información necesaria para formular un problema de ruteo de vehículos con ventanas de tiempo y capacidades múltiples.

Para cada orden se dispone, entre otros, de los siguientes atributos relevantes:

- Identificador de cliente
- Demanda en peso (WEIGHT)
- Demanda en volumen (VOLUME)
- Tiempo de servicio (SERVICE_TIME)
- Ventana de tiempo original (TIME_WINDOW)

Adicionalmente, para cada día se cuenta con matrices de tiempo de viaje entre todos los nodos (depósito y clientes), generadas bajo tres supuestos distintos de condiciones operacionales: Escenario optimista, escenario más probable y escenario pesimista. Estas matrices permiten modelar explícitamente la incertidumbre en los tiempos de viaje y son posteriormente integradas mediante una distribución PERT-Beta.

Las capacidades de los vehículos se definen de forma homogénea para toda la flota, considerando simultáneamente restricciones de peso y volumen, lo que da lugar a un problema de ruteo con múltiples capacidades.

Generación de escenarios de ventanas de tiempo

Con el objetivo de evaluar el desempeño de los métodos de resolución bajo distintos niveles de dificultad operacional, las ventanas de tiempo originales son reemplazadas por ventanas sintéticas generadas de forma controlada. Este proceso permite crear escenarios con distintos grados de holgura temporal y solapamiento entre clientes.
Se definen cuatro tipos de escenarios, caracterizados principalmente por la duración de las ventanas y el grado de superposición temporal:

- Ultra tight: ventanas muy estrechas y altamente solapadas
- Tight: ventanas estrechas con solapamiento moderado
- Medium: ventanas de duración intermedia
- Loose: ventanas amplias con bajo solapamiento

Para cada escenario, se aplican tres estrategias de generación distintas:

- Estrategia aleatoria (random): Las ventanas de tiempo se generan de manera independiente para cada cliente. La duración de la ventana se muestrea dentro del rango definido por el escenario y se asegura que sea mayor que el tiempo de servicio del cliente. El inicio de la ventana se selecciona uniformemente dentro del horizonte temporal disponible. Esta estrategia produce escenarios sin estructura temporal explícita y sirve como línea base de comparación.

- Estrategia agrupada (clustered): Los clientes se asignan aleatoriamente a un número fijo de clústeres temporales. Cada clúster posee un centro temporal y las ventanas de los clientes se generan alrededor de dicho centro utilizando una distribución normal acotada. Esta estrategia introduce concentraciones temporales de clientes, simulando escenarios donde múltiples entregas deben realizarse en franjas horarias similares.
- Estrategia encadenada (chained): Los clientes se agrupan en secuencias asociadas a vehículos mínimos requeridos según las capacidades de peso y volumen. Dentro de cada grupo, las ventanas de tiempo se generan de forma secuencial, de modo que cada ventana se superpone parcialmente con la anterior según un ratio de solapamiento aleatorio. Este enfoque produce escenarios altamente estructurados, donde existe una dependencia temporal explícita entre clientes, emulando rutas casi forzadas y de alta dificultad.

Para cada día, escenario y estrategia, se genera un archivo independiente en formato CSV, lo que permite una evaluación sistemática y reproducible de los métodos de resolución.

Preparación del modelo de datos para los solvers

Los escenarios generados se transforman en una estructura de datos común utilizada por todos los solvers (deterministas, heurísticos y basados en aprendizaje por refuerzo). Este proceso se realiza mediante una función de creación de modelo que encapsula todos los parámetros del problema.

En primer lugar, los tiempos de viaje se integran utilizando una distribución PERT-Beta, combinando las matrices optimista, más probable y pesimista. Dependiendo del modo de operación, se utiliza el valor esperado de la distribución (caso determinista) o una muestra aleatoria controlada por semilla (caso estocástico).

Posteriormente, se construyen los siguientes elementos del modelo:

- Matriz de tiempos efectiva, incorporando la incertidumbre
- Ventanas de tiempo, incluyendo el depósito
- Demandas de peso y volumen
- Tiempos de servicio
- Capacidades de los vehículos
- Número máximo de vehículos disponibles
- Número mínimo de vehículos requeridos, calculado a partir de las demandas agregadas
- Horizonte temporal del problema

Adicionalmente, se calculan estadísticas globales de las ventanas de tiempo (media, desviación estándar, mínimo y máximo), las cuales son utilizadas tanto para análisis exploratorio como para la construcción de estados en el entorno de aprendizaje por refuerzo.

Finalmente, el modelo incluye una función de costo ponderada que considera: tiempo total de viaje, tardanzas, penalización por uso de vehículos adicionales y costo fijo por vehículo utilizado.

Esta representación unificada permite que distintos métodos de resolución operen sobre exactamente la misma instancia del problema, garantizando comparabilidad y consistencia experimental.

4.2. Metodología general

El problema abordado en este trabajo corresponde a una variante del Problema de Ruteo de Vehículos con Ventanas de Tiempo y Capacidad (CVRPTW), extendido para considerar incertidumbre operacional en variables críticas del proceso logístico, tales como los tiempos de viaje y de servicio. En este contexto, la planificación de rutas debe no sólo cumplir con las restricciones clásicas del problema, sino también ofrecer soluciones robustas frente a la variabilidad inherente del entorno, minimizando el impacto de retrasos y riesgos operacionales. Para ello, se adopta un enfoque metodológico progresivo que combina métodos deterministas, heurísticas avanzadas y aprendizaje por refuerzo, evaluados bajo un marco común de simulación estocástica.

La metodología propuesta se estructura como un flujo escalonado de modelos, donde cada etapa cumple un rol específico tanto en la generación de soluciones como en la comparación de desempeño. En primer lugar, se construyen baselines deterministas y heurísticos que permiten establecer referencias de calidad y costo computacional. Posteriormente, se introduce un modelo de aprendizaje por refuerzo diseñado para mejorar el proceso de búsqueda mediante la selección adaptativa de operadores dentro de un esquema ALNS. Finalmente, el modelo entrenado se integra en el proceso de resolución del problema para obtener soluciones guiadas por políticas aprendidas.

El primer método considerado corresponde a un baseline basado en OR-Tools. Este enfoque utiliza un solver determinista para generar un conjunto inicial de rutas factibles, respetando las restricciones de capacidad y ventanas de tiempo. Su principal objetivo

dentro de la metodología es servir como punto de comparación inicial, proporcionando soluciones obtenidas mediante técnicas clásicas de optimización. Este baseline permite cuantificar el desempeño de un enfoque tradicional frente a métodos más flexibles bajo condiciones de incertidumbre.

En segundo lugar, se implementa un baseline basado en Adaptive Large Neighborhood Search (ALNS). Este método heurístico explora el espacio de soluciones mediante la aplicación iterativa de operadores de destrucción y reparación, permitiendo escapar de óptimos locales y adaptarse mejor a problemas de gran complejidad. En este baseline, la selección de operadores se realiza de forma no guiada por aprendizaje, incorporando mecanismos de aceptación basados en enfriamiento tipo Simulated Annealing. Este modelo introduce reglas adicionales de factibilidad y penalización que no están presentes en el enfoque determinista, con el fin de reflejar de manera más realista las condiciones operativas del problema.

A continuación, se desarrolla un modelo entrenado mediante aprendizaje por refuerzo cuyo objetivo es aprender una política de selección de operadores ALNS. En este esquema, el proceso de búsqueda se formula como un problema de decisión secuencial, donde el agente observa el estado actual de la solución y decide qué combinación de operadores aplicar en cada iteración. El entrenamiento se realiza sobre múltiples instancias y escenarios, permitiendo que el agente aprenda estrategias que balancean exploración, explotación y costo computacional, con el fin de mejorar de manera consistente la calidad de las soluciones generadas.

Finalmente, el modelo entrenado se utiliza para resolver el problema de ruteo mediante un ALNS guiado por aprendizaje por refuerzo. En esta etapa, la política aprendida reemplaza los mecanismos heurísticos tradicionales de selección de operadores, orientando la búsqueda hacia regiones del espacio de soluciones con mejor desempeño esperado bajo incertidumbre. Este enfoque constituye la propuesta principal del trabajo y es evaluado de manera directa frente a los baselines previamente descritos.

La comparación entre los distintos métodos se realiza bajo un marco común de evaluación basado en simulación Monte Carlo y generación de escenarios estocásticos. Cada solución obtenida es evaluada múltiples veces considerando realizaciones aleatorias de las variables inciertas, lo que permite estimar métricas de desempeño robustas. Las métricas utilizadas incluyen la tardanza total y promedio, el riesgo medido mediante el Conditional Value at Risk (CVaR) a 95%, los tiempos totales de viaje, el número de vehículos utilizados y el tiempo computacional requerido por cada método. Este enfoque garantiza una comparación justa y consistente, permitiendo analizar no sólo la calidad promedio de las soluciones, sino también su comportamiento en escenarios adversos y su viabilidad computacional en contextos reales.

4.3. Baseline determinista con OR-Tools

Como primer punto de referencia metodológico, se utiliza un baseline determinista basado en OR-Tools, cuyo rol principal es generar un conjunto de soluciones iniciales factibles que sirvan como base de comparación para los métodos heurísticos y de aprendizaje posteriores. Este baseline no busca capturar toda la complejidad del

problema bajo incertidumbre, sino establecer un estándar controlado y reproducible de desempeño, sobre el cual se evalúan métricas de tardanza, riesgo y tiempos de viaje una vez que las soluciones son sometidas a escenarios estocásticos.

El solver implementado utiliza el routing solver de OR-Tools para resolver una formulación del VRPTW en la que el objetivo principal está alineado con el desempeño temporal de las rutas. El costo de los arcos se define como la suma del tiempo de viaje y el tiempo de servicio en el nodo de origen, lo que permite que la función objetivo represente de forma directa el tiempo efectivo consumido por la operación. La tardanza respecto a las ventanas de tiempo se permite explícitamente y se incorpora al objetivo mediante penalizaciones suaves (soft upper bounds) sobre la dimensión temporal, de modo que las soluciones tardías no se descartan, pero sí se castigan fuertemente.

El tiempo de resolución del solver se fija de manera estricta en 60 segundos. Esta decisión se fundamenta en observaciones empíricas realizadas durante la etapa experimental, donde se constató que aumentar el tiempo disponible para la búsqueda no produce mejoras significativas en la calidad de las soluciones obtenidas, ni en términos de tardanza ni de tiempos de viaje. En consecuencia, mantener un tiempo fijo permite una comparación más justa con otros métodos y asegura una carga computacional acotada y consistente.

Un aspecto relevante de este baseline es que la cantidad de vehículos no constituye una variable de decisión del solver. El número de vehículos se fija de antemano al mínimo necesario para satisfacer las restricciones de capacidad definidas en los datos de entrada. De esta forma, OR-Tools no explora soluciones con distintos tamaños de flota, y el uso

de vehículos queda completamente determinado antes de la resolución. Este diseño refuerza el carácter determinista del baseline y evita que el solver reduzca artificialmente la tardanza o el costo total mediante el uso de vehículos adicionales.

En cuanto a los métodos de búsqueda, OR-Tools emplea una estrategia de construcción inicial basada en Parallel Cheapest Insertion, que suele presentar un buen desempeño en problemas con ventanas de tiempo. Posteriormente, la solución es refinada mediante una metaheurística de búsqueda local del tipo Guided Local Search, la cual permite reorganizar las rutas y escapar de óptimos locales penalizando patrones de arcos frecuentemente utilizados. Adicionalmente, se habilita la propagación completa de restricciones, lo que mejora la consistencia del modelo temporal durante la búsqueda.

El modelo incluye una dimensión temporal con holgura suficiente para permitir esperas automáticas cuando un vehículo llega antes del inicio de la ventana de tiempo, evitando así infactibilidad de la solución. Asimismo, se incorpora una penalización extremadamente alta para la exclusión de clientes mediante disyunciones, con el objetivo de forzar que todas las demandas sean atendidas y evitar soluciones degeneradas en las que el solver elimine nodos para reducir la tardanza.

A partir de las soluciones generadas por OR-Tools se extraen las rutas y los tiempos reales de llegada a cada nodo, los cuales se utilizan posteriormente en el marco de evaluación estocástico. Sobre estas soluciones deterministas se calculan métricas de tardanza total, tiempos totales de viaje y CVAR95, todas ellas obtenidas mediante simulación Monte Carlo bajo distintos escenarios de incertidumbre. De este modo,

aunque OR-Tools opera de forma determinista, sus soluciones son evaluadas bajo el mismo esquema probabilístico que los métodos más avanzados.

Finalmente, es importante destacar que este baseline no modela explícitamente varias restricciones y lógicas que sí se incorporan en los enfoques posteriores. En particular, no se consideran capacidades múltiples, reglas operacionales adicionales ni mecanismos adaptativos de redistribución de carga más allá de lo que permite la formulación estándar del routing solver. Estas simplificaciones son intencionales y refuerzan el carácter del modelo como línea base, cuyo propósito es servir como punto de comparación inicial frente a métodos heurísticos y de aprendizaje más expresivos y flexibles.

4.4. Baseline Adaptive Large Neighborhood Search (ALNS)

El segundo enfoque considerado corresponde a un baseline estocástico basado en Adaptive Large Neighborhood Search (ALNS). Este método se utiliza como referencia heurística avanzada frente al baseline determinista, permitiendo explorar de forma más flexible el espacio de soluciones del problema de ruteo bajo múltiples restricciones. A diferencia de OR-Tools, ALNS no resuelve una formulación matemática cerrada, sino que itera sobre soluciones completas aplicando operadores que destruyen y reparan parcialmente las rutas, adaptándose progresivamente a la estructura del problema.

El rol del ALNS dentro de la metodología es doble. Por una parte, actúa como un método robusto capaz de manejar explícitamente restricciones operacionales más complejas que el baseline determinista. Por otra, establece una línea base directa para evaluar el impacto de incorporar aprendizaje por refuerzo en la selección de operadores,

ya que el mismo framework puede operar tanto en modo aleatorio como guiado por una política aprendida.

Estructura general del algoritmo ALNS

El algoritmo comienza a partir de una solución inicial factible, construida mediante un procedimiento greedy de inserción que asigna los clientes a un número fijo mínimo de vehículos. A partir de esta solución, el ALNS ejecuta un proceso iterativo compuesto por cuatro etapas principales: selección de operadores, destrucción parcial de la solución, reparación de la solución resultante y decisión de aceptación basada en Simulated Annealing. Este ciclo se repite durante un número máximo de iteraciones.
En este baseline, la selección de operadores se realiza de forma aleatoria uniforme sobre el conjunto de combinaciones posibles entre operadores de destrucción y reparación. No se utilizan pesos adaptativos ni aprendizaje en esta etapa, de modo que el comportamiento del algoritmo es puramente estocástico, controlado únicamente por la lógica de aceptación y el esquema de enfriamiento.

Operadores de destrucción

Los operadores de destrucción tienen como objetivo remover parcial y estratégicamente clientes de las rutas actuales, generando espacios para una posterior reconstrucción potencialmente mejor. En el baseline implementado se utilizan los siguientes tipos de operadores basados en (Wouda, s.f.):

- Destrucción aleatoria (random destroy): elimina una fracción fija de clientes seleccionados al azar desde todas las rutas, con distintos niveles de intensidad.

Este operador favorece la exploración global del espacio de soluciones y permite romper estructuras rígidas de rutas consolidadas.

- Reubicación (relocate destroy): selecciona un cliente de una ruta y lo mueve a otra ruta distinta en una posición aleatoria. Este operador introduce cambios más locales, enfocados en redistribuir carga entre vehículos.
- Destrucción por peor contribución (worst destroy): identifica clientes cuya contribución marginal al tiempo de viaje es mayor y los elimina prioritariamente. Este operador es más dirigido, ya que intenta atacar explícitamente componentes costosos de la solución.

Durante la destrucción, los clientes removidos se almacenan como no servidos, quedando pendientes de reinserción durante la fase de reparación.

Operadores de reparación

Los operadores de reparación reconstruyen las soluciones incompletas reintegrando los clientes removidos, buscando mantener la factibilidad y mejorar el costo total. En este baseline se consideran los siguientes operadores basado en (Wouda, s.f.):

- Reparación greedy: reintroduce clientes no servidos en la primera posición factible encontrada, priorizando la cobertura completa y la factibilidad sobre la optimalidad local.
- Reparación orientada a ventanas de tiempo: detecta rutas que violan restricciones temporales o de capacidad, elimina clientes problemáticos y los reinsertan

posteriormente mediante un proceso greedy. Este operador es especialmente relevante en escenarios con ventanas estrechas.

- Reparación tipo regret-2: evalúa, para cada cliente no servido, las dos mejores inserciones posibles y selecciona aquel con mayor valor de arrepentimiento (regret). Este enfoque prioriza clientes difíciles de insertar y suele producir soluciones más balanceadas. Adicionalmente, se incorpora una heurística específica para asegurar que se utilice al menos el número mínimo de vehículos requerido, redistribuyendo clientes desde rutas sobrecargadas hacia rutas vacías cuando sea necesario.

Uso de Simulated Annealing

El criterio de aceptación de soluciones en el ALNS se basa en un esquema clásico de Simulated Annealing. En cada iteración, la solución candidata generada tras aplicar los operadores es comparada con la solución actual. Si la nueva solución presenta un menor costo, es aceptada de forma determinista. En caso contrario, puede ser aceptada con una probabilidad positiva que depende del incremento de costo y de la temperatura actual del sistema.

La temperatura inicial se fija en un valor alto, permitiendo una mayor probabilidad de aceptar soluciones peores durante las primeras iteraciones, lo que favorece la exploración del espacio de soluciones. Posteriormente, la temperatura se reduce de forma multiplicativa mediante un factor de enfriamiento constante, haciendo que el algoritmo sea progresivamente más selectivo y se concentre en la explotación de buenas

soluciones. Este mecanismo permite escapar de óptimos locales en etapas tempranas y estabilizar la búsqueda hacia el final del proceso.

Evaluación de soluciones y manejo de factibilidad

La evaluación de soluciones en el ALNS se realiza mediante una función de costo explícita que combina múltiples componentes relevantes para el problema. El costo total incluye el tiempo total de viaje, la tardanza acumulada respecto a las ventanas de tiempo, uso de vehículos adicionales sobre el mínimo requerido y costos fijos por vehículo utilizado.

A diferencia del baseline con OR-Tools, el ALNS modela explícitamente restricciones de capacidad tanto de peso como de volumen, tiempos de servicio, ventanas de tiempo con espera gratuita y penalización por tardanza, así como restricciones duras sobre el número mínimo de vehículos utilizados. Cualquier violación de capacidad o de uso mínimo de vehículos resulta en una solución infactible, penalizada con un costo infinito.

En conjunto, este baseline con ALNS representa un método heurístico robusto y expresivo, capaz de capturar restricciones y lógicas operacionales que no se consideran en el enfoque determinista. Su desempeño sirve como referencia directa para evaluar el beneficio adicional de introducir aprendizaje por refuerzo en la selección adaptativa de operadores, manteniendo constante el resto del framework de búsqueda.

Cuantificación del esfuerzo computacional mediante ticks de operadores ALNS

Con el objetivo de comparar de manera homogénea el esfuerzo computacional entre distintas variantes del framework ALNS incluyendo su versión clásica y aquella guiada

por aprendizaje por refuerzo se introduce una métrica abstracta denominada ticks. Esta métrica no representa tiempo de cómputo real, sino una aproximación estructural al costo computacional relativo asociado a la aplicación de cada operador de destrucción y reparación.

Un tick corresponde a una unidad elemental de esfuerzo algorítmico, asociada a operaciones básicas tales como recorridos de rutas, evaluaciones de factibilidad, cálculos de costos marginales, inserciones de clientes o verificaciones de restricciones. La contabilización de ticks se realiza explícitamente dentro de cada operador, acumulándose a medida que el algoritmo ejecuta bucles, evaluaciones y transformaciones sobre las soluciones.

En los operadores de destrucción, los ticks reflejan principalmente:

- El recorrido de las rutas completas para identificar clientes candidatos a remover.
- El cálculo de contribuciones marginales al tiempo de viaje, en el caso del operador worst destroy.
- El costo asociado a la manipulación de estructuras de datos, como la eliminación y relocalización de clientes entre rutas.

Por ejemplo, el operador random destroy acumula ticks proporcionales al tamaño de las rutas inspeccionadas y al número total de clientes considerados para remoción, mientras que el operador worst destroy incorpora un costo adicional asociado a la evaluación explícita del impacto marginal de cada cliente y al ordenamiento de dichas contribuciones.

En los operadores de reparación, los ticks capturan:

- El número de intentos de inserción evaluados para cada cliente no servido.
- Las verificaciones de factibilidad de ruta, incluyendo restricciones de capacidad, ventanas de tiempo y tiempos de servicio.
- Los cálculos de costo incremental necesarios para comparar distintas posiciones de inserción, particularmente relevantes en el operador regret-2.

Operadores más sofisticados, como repair_time_windows y repair_regret2, presentan una mayor acumulación de ticks debido a la combinación de ciclos internos, evaluaciones repetidas de factibilidad y recomputaciones parciales de costos de ruta.

El número total de ticks de una iteración del ALNS se obtiene como la suma de los ticks generados por el operador de destrucción seleccionado y el operador de reparación aplicado posteriormente. A lo largo de la ejecución completa del algoritmo, los ticks se acumulan de forma aditiva, permitiendo medir el esfuerzo computacional total invertido en la búsqueda.

Esta métrica resulta especialmente útil para el análisis comparativo entre métodos, ya que permite desacoplar el esfuerzo algorítmico del tiempo de ejecución dependiente de hardware, implementación o paralelismo. En particular, los ticks constituyen una base común para analizar la eficiencia relativa entre el ALNS clásico con selección aleatoria de operadores y el enfoque PPO-ALNS, donde una política aprendida tiende a priorizar operadores con mejor relación entre mejora de solución y costo computacional.

4.5. Modelo de aprendizaje por refuerzo para selección de operadores ALNS

Con el objetivo de superar las limitaciones de la selección heurística de operadores en el ALNS clásico, se propone un modelo de aprendizaje por refuerzo que aprende de forma explícita una política para seleccionar operadores de destrucción y reparación. Este modelo no reemplaza la lógica de búsqueda del ALNS, sino que se integra directamente en su núcleo, sustituyendo la selección aleatoria de operadores por decisiones guiadas por una política entrenada. De esta forma, se mantiene la estructura del ALNS, pero se introduce adaptabilidad basada en experiencia acumulada sobre múltiples escenarios.

Formulación del entorno de aprendizaje

El problema se formula como un entorno de aprendizaje por refuerzo episódico, implementado siguiendo la interfaz estándar de OpenAI Gym. Cada episodio corresponde a la resolución de una instancia del problema de ruteo generada a partir de un día, escenario y método de construcción específicos, muestreados de forma cíclica durante el entrenamiento. El entorno encapsula completamente la dinámica interna del ALNS, permitiendo que el agente interactúe únicamente a través de la selección de operadores.

El estado del entorno se representa mediante un vector de observaciones de dimensión fija, diseñado para capturar tanto el estado de la búsqueda como características estructurales del escenario. Las observaciones incluyen la razón entre el costo de la solución actual y el mejor costo observado, el uso relativo de vehículos respecto al

mínimo requerido, el exceso de vehículos, la temperatura normalizada del esquema de Simulated Annealing, el progreso relativo del episodio, la última acción ejecutada junto con su recompensa asociada y estadísticas normalizadas de las ventanas de tiempo del escenario (media, desviación estándar, mínimo y máximo). Este conjunto de variables proporciona al agente información suficiente para inferir el estado de la exploración y la dificultad del problema.

El espacio de acciones es discreto y corresponde a la selección de un par operador de destrucción–operador de reparación. Cada acción representa una combinación específica de operadores ALNS previamente definidos, de modo que el agente aprende directamente qué tipo de perturbación y reconstrucción aplicar en cada iteración. Las transiciones del entorno se producen al ejecutar una iteración completa del ALNS: la solución actual es modificada por los operadores seleccionados, evaluada y, dependiendo del criterio de aceptación, se actualiza o no el estado del sistema.

Definición de la función de recompensa

La función de recompensa está diseñada para incentivar mejoras locales en la calidad de las soluciones, al tiempo que penaliza comportamientos computacionalmente costosos o estructuralmente indeseables. El componente principal de la recompensa corresponde a la mejora relativa del costo de la solución tras aplicar una acción, normalizada respecto al costo base. De este modo, el agente es recompensado cuando selecciona operadores que conducen a soluciones de menor costo.

Adicionalmente, se incorpora una penalización asociada al esfuerzo computacional de los operadores, aproximado mediante el número de ticks consumidos durante las fases

de destrucción y reparación. Esta penalización se aplica únicamente cuando la acción no produce una mejora, evitando desincentivar operadores costosos que resultan efectivos. También se incluye una penalización suave por el uso de vehículos adicionales por sobre el mínimo requerido, alineando el aprendizaje con los objetivos operacionales del problema.

La recompensa total es finalmente transformada mediante una función hiperbólica tangente, lo que limita su magnitud y estabiliza el proceso de entrenamiento. Este diseño permite balancear exploración y explotación, evitando recompensas extremas que puedan desestabilizar el aprendizaje.

Dinámica de episodios y pasos de entrenamiento

Cada episodio comienza con la carga de un nuevo escenario de entrenamiento y la construcción de una solución inicial factible mediante el mismo procedimiento utilizado en el ALNS baseline. La temperatura del esquema de Simulated Annealing se inicializa en un valor alto y se reduce progresivamente en cada paso según una tasa de enfriamiento fija, replicando la dinámica de aceptación estocástica del ALNS clásico. Dentro de cada episodio, el agente interactúa con el entorno durante un número máximo de iteraciones internas, que actúan como condición de término del episodio. En cada iteración, el agente selecciona una acción, se ejecuta una iteración de ALNS con aceptación por Simulated Annealing, se calcula la recompensa y se actualiza el estado del entorno. El mejor estado alcanzado durante el episodio se mantiene de forma separada, aunque la evolución del episodio continúa a partir de la solución corriente.

El entrenamiento se realiza sobre múltiples episodios consecutivos, abarcando diferentes días y configuraciones de escenarios, lo que permite al agente aprender una política robusta frente a variabilidad estructural y operativa del problema.

Justificación del uso de Proximal Policy Optimization (PPO)

Para el entrenamiento del agente se utiliza el algoritmo Proximal Policy Optimization (PPO), una técnica de aprendizaje por refuerzo de política que ha demostrado buen desempeño y estabilidad en entornos con espacios de observación continuos y acciones discretas. PPO ofrece un equilibrio adecuado entre simplicidad de implementación y robustez del aprendizaje, gracias a su mecanismo de actualización restringida que evita cambios abruptos en la política (OpenAI, s.f.).

Dado que el entorno presenta recompensas ruidosas, transiciones estocásticas y dependencias temporales largas propias de la dinámica del ALNS, PPO se presenta como una opción apropiada para aprender políticas estables sin requerir una afinación excesiva de hiperparámetros. Asimismo, su compatibilidad con normalización de observaciones y recompensas facilita el entrenamiento en escenarios heterogéneos, reforzando la generalización del modelo entrenado.

En conjunto, este modelo de aprendizaje por refuerzo permite reemplazar la selección heurística de operadores por una política adaptativa, manteniendo la estructura del ALNS y potenciando su capacidad de explorar eficientemente el espacio de soluciones bajo incertidumbre.

4.6. Resolución del problema con ALNS guiado por aprendizaje por refuerzo

La etapa final de la metodología corresponde a la resolución del problema de ruteo utilizando un esquema de ALNS guiado por aprendizaje por refuerzo. En este enfoque, el modelo entrenado se integra directamente en el flujo del ALNS para tomar decisiones informadas sobre la selección de operadores en cada iteración, manteniendo intacta la estructura general del algoritmo y sus mecanismos de evaluación y aceptación.

El flujo de resolución comienza con la construcción de una solución inicial factible, utilizando el mismo procedimiento greedy definido para el baseline ALNS. A partir de esta solución, se ejecuta un ciclo iterativo compuesto por selección de operadores, aplicación de destrucción y reparación, evaluación de la solución candidata y aceptación mediante Simulated Annealing. Este ciclo se repite durante un número fijo de iteraciones, con una temperatura inicial elevada que se reduce progresivamente siguiendo el mismo esquema de enfriamiento que en el baseline heurístico.

La integración del modelo de aprendizaje por refuerzo se produce exclusivamente en la etapa de selección de operadores. En lugar de escoger de forma aleatoria una combinación de operador de destrucción y reparación, el estado actual del proceso ALNS se transforma en un vector de observación consistente con el utilizado durante el entrenamiento del agente. Este vector se entrega al modelo entrenado, el cual infiere de manera determinista la acción a ejecutar, es decir, el par de operadores que se considera

más adecuado para el estado actual de la búsqueda. De esta forma, la política aprendida reemplaza la heurística de selección, sin alterar el resto del flujo del algoritmo.

En comparación con el baseline ALNS, la principal diferencia radica en la forma en que se decide qué operadores aplicar en cada iteración. Mientras que el enfoque clásico explora el espacio de combinaciones de manera uniforme y no informada, el ALNS guiado por aprendizaje por refuerzo introduce una adaptación dinámica basada en la experiencia previa del agente. La política entrenada es capaz de condicionar sus decisiones al estado de la búsqueda, al progreso del algoritmo, a la temperatura del esquema de Simulated Annealing y a características estructurales del escenario, favoreciendo operadores que históricamente han producido mejoras en contextos similares.

Esta adaptación dinámica tiene un impacto esperado tanto en la calidad como en la estabilidad de las soluciones. Al priorizar operadores que tienden a generar mejoras consistentes, el algoritmo puede concentrar su esfuerzo computacional en regiones más prometedoras del espacio de soluciones, reduciendo la variabilidad entre ejecuciones y mejorando el desempeño promedio bajo incertidumbre. Asimismo, la política aprendida puede modular implícitamente el balance entre exploración y explotación, aprovechando mejor las fases tempranas de alta temperatura y siendo más selectiva a medida que el proceso converge.

Es importante destacar que, salvo por la selección de operadores, todas las demás componentes del flujo se mantienen idénticas al baseline ALNS. La evaluación de soluciones, el manejo de factibilidad, las penalizaciones por tardanza, uso de vehículos y

violaciones de capacidad, así como el criterio de aceptación por Simulated Annealing y el esquema de enfriamiento, permanecen sin modificaciones. Esta consistencia metodológica permite atribuir de forma directa cualquier diferencia de desempeño observada al efecto del aprendizaje por refuerzo, aislando su impacto sobre el proceso de búsqueda.

En síntesis, el ALNS guiado por aprendizaje por refuerzo representa una extensión directa del baseline heurístico, donde el conocimiento adquirido durante el entrenamiento se utiliza para orientar la exploración del espacio de soluciones. Este enfoque combina la flexibilidad del ALNS con la capacidad adaptativa del aprendizaje por refuerzo, constituyendo el método final propuesto para la resolución del problema de ruteo bajo incertidumbre.

4.7. Parámetros de los métodos y justificación

En esta sección se describen los principales parámetros utilizados en cada uno de los métodos evaluados, junto con la justificación técnica de su selección. En todos los casos, la elección de parámetros busca un equilibrio explícito entre calidad de las soluciones obtenidas, estabilidad del desempeño bajo incertidumbre y costo computacional, de manera coherente con un contexto de aplicación realista.

Parámetros del baseline determinista con OR-Tools

El baseline determinista basado en OR-Tools utiliza una configuración deliberadamente acotada y estable, orientada a generar soluciones base reproducibles.

- Tiempo máximo de cómputo: Se fija un límite de 60 segundos por instancia. Este valor se selecciona a partir de observaciones empíricas que muestran que incrementos adicionales en el tiempo de búsqueda no producen mejoras relevantes en la calidad de las soluciones, especialmente en términos de tardanza y tiempos de viaje.
- Número de vehículos: La cantidad de vehículos se fija al mínimo necesario para cumplir las restricciones de capacidad, calculado previamente a la resolución. No se permite que el solver decida el tamaño de la flota, evitando que la mejora de métricas se logre artificialmente mediante el uso de vehículos adicionales.
- Estrategia de búsqueda: Se utiliza Parallel Cheapest Insertion como heurística de solución inicial y Guided Local Search como metaheurística de mejora. Esta combinación es adecuada para problemas con ventanas de tiempo y ofrece un compromiso razonable entre rapidez y calidad.

Esta parametrización convierte a OR-Tools en una línea base sólida pero deliberadamente limitada en expresividad, adecuada para comparar con enfoques heurísticos más flexibles.

Parámetros del baseline ALNS

El baseline con ALNS introduce mayor libertad exploratoria y, en consecuencia, requiere una parametrización que controle tanto la calidad de búsqueda como el esfuerzo computacional.

- Número máximo de iteraciones: Se fija en 3.000 iteraciones internas. Este valor permite observar convergencia del algoritmo sin incurrir en tiempos de cómputo excesivos, manteniendo la comparabilidad con el enfoque guiado por aprendizaje.
- Temperatura inicial: Se utiliza una temperatura inicial alta (1.000), lo que favorece la aceptación de soluciones peores durante las primeras etapas de la búsqueda y permite escapar de óptimos locales tempranos.
- Esquema de enfriamiento: Se adopta un enfriamiento multiplicativo con una tasa de 0,998. Este valor asegura una disminución gradual de la temperatura, manteniendo exploración suficiente durante una fracción relevante de las iteraciones antes de concentrarse en explotación.
- Selección de operadores: Los operadores de destrucción y reparación se seleccionan de forma aleatoria uniforme entre todas las combinaciones disponibles. Esta decisión define el carácter puramente estocástico del baseline y establece un punto de comparación directo con el modelo guiado por aprendizaje.

El conjunto de parámetros del ALNS baseline busca maximizar la diversidad de soluciones exploradas, aceptando un mayor costo computacional a cambio de una mayor capacidad de adaptación estructural frente a OR-Tools.

Parámetros del ALNS guiado por aprendizaje por refuerzo

El ALNS con aprendizaje por refuerzo mantiene los parámetros estructurales del baseline ALNS, modificando únicamente aquellos relacionados con la toma de decisiones y el entrenamiento del agente.

- Iteraciones internas y enfriamiento: Se conservan el número máximo de iteraciones (3.000), la temperatura inicial (1.000) y la tasa de enfriamiento (0,998). Esta decisión permite aislar el impacto del aprendizaje por refuerzo, asegurando que cualquier diferencia de desempeño se deba exclusivamente a la selección de operadores.
- Algoritmo de aprendizaje: Se utiliza Proximal Policy Optimization (PPO) con una política tipo MLP. PPO se selecciona por su estabilidad frente a recompensas ruidosas y entornos estocásticos, características inherentes al proceso ALNS.
- Parámetros de entrenamiento del modelo RL: El entrenamiento se realiza durante 300.000 timesteps, con una tasa de aprendizaje de 10^{-4}, factor de descuento γ=0.99, tamaño de batch de 256 y ventanas de actualización de 2.048 pasos. Estos valores fueron determinados en base a ensayo y error y representan un compromiso entre estabilidad del aprendizaje y tiempo total de entrenamiento.
- Normalización: Se emplea normalización de observaciones y recompensas, junto con un recorte suave de recompensas, para facilitar la generalización del agente a distintos días y configuraciones de escenarios.
- Selección de operadores en inferencia: Durante la resolución final, la política entrenada se utiliza de forma determinista para seleccionar operadores,

priorizando estabilidad y reproducibilidad de resultados por sobre exploración adicional.

Consideraciones de costo computacional y calidad de solución

En todos los métodos, los parámetros se seleccionan considerando explícitamente el equilibrio entre calidad de solución y costo computacional. OR-Tools prioriza rapidez y reproducibilidad, el ALNS baseline maximiza flexibilidad exploratoria a un costo computacional mayor, y el ALNS guiado por aprendizaje por refuerzo busca mejorar la eficiencia de dicha exploración, concentrando el esfuerzo computacional en decisiones con mayor probabilidad de mejora.

Esta coherencia en la parametrización permite realizar comparaciones justas entre métodos y evaluar de manera aislada el aporte del aprendizaje por refuerzo en un marco común de resolución y evaluación bajo incertidumbre.

4.8. Metodología de evaluación mediante simulación Monte Carlo y CVaR95

La evaluación de los métodos propuestos se realiza bajo un enfoque explícito de incertidumbre, reconociendo que las soluciones de ruteo obtenidas por los distintos algoritmos corresponden a planes deterministas que, en la práctica, se ejecutan en entornos estocásticos. En este contexto, la calidad de una solución no se mide únicamente por su desempeño esperado, sino también por su comportamiento frente a escenarios adversos. Para capturar esta dimensión de riesgo, se utiliza una metodología

de evaluación basada en simulación Monte Carlo y métricas de riesgo, con énfasis en el CVaR95.

Enfoque general de evaluación bajo incertidumbre

El proceso de evaluación se estructura de manera uniforme para todos los métodos considerados. Cada algoritmo genera una solución base determinista, definida por un conjunto fijo de rutas. Estas rutas no se modifican durante la evaluación; en cambio, se simula su ejecución repetidas veces bajo distintas realizaciones de las variables aleatorias del sistema. De esta forma, se aísla el efecto de la incertidumbre operacional sobre un plan de ruteo específico, permitiendo comparar soluciones en términos de robustez y riesgo.

Este enfoque asegura que las diferencias observadas en las métricas de desempeño se deban exclusivamente a la calidad estructural de las soluciones generadas por cada método, y no a variaciones en la lógica de simulación o evaluación.

Simulación Monte Carlo de escenarios estocásticos

La simulación Monte Carlo se utiliza para generar múltiples escenarios de ejecución a partir de una solución fija. Para cada conjunto de rutas, se generan 1.000 escenarios independientes, en los cuales se simulan los tiempos efectivos de viaje entre nodos. La simulación se realiza utilizando semillas controladas, lo que garantiza la reproducibilidad de los resultados y permite comparaciones justas entre métodos bajo exactamente las mismas realizaciones estocásticas.

En cada escenario, las rutas se recorren secuencialmente aplicando una lógica estándar de CVRPTW: se considera espera automática cuando un vehículo llega antes del inicio de la ventana de tiempo, y se calcula la tardanza cuando el inicio efectivo de servicio excede el límite superior de la ventana. A partir de esta simulación se obtienen métricas agregadas de desempeño para cada escenario, tales como la tardanza total, el porcentaje de clientes atendidos a tiempo y el tiempo total de viaje.

Variables aleatorias consideradas

La incertidumbre modelada en la evaluación se concentra en los tiempos de viaje entre nodos. Estos tiempos se generan de forma estocástica a partir de distribuciones tipo Beta-PERT, parametrizadas mediante estimaciones optimista, más probable y pesimista. Para cada escenario de Monte Carlo, se genera una matriz completa de tiempos de viaje, consistente para todas las rutas evaluadas en dicho escenario.

Los tiempos de servicio y las ventanas de tiempo se mantienen fijos durante la simulación, de modo que la variabilidad del desempeño proviene exclusivamente de la incertidumbre en los desplazamientos. Esta decisión permite analizar de forma clara el impacto de la variabilidad en los tiempos de viaje sobre la tardanza y el cumplimiento de ventanas, sin introducir fuentes adicionales de ruido.

Métricas estadísticas de desempeño

A partir de los resultados de la simulación Monte Carlo se construye una distribución empírica de la tardanza total asociada a cada solución. Sobre esta distribución se calculan varias métricas estadísticas relevantes. La tardanza esperada corresponde al

promedio de la tardanza total sobre todos los escenarios simulados y representa el desempeño medio de la solución. La variabilidad de la tardanza se captura mediante su desviación estándar, que cuantifica la dispersión de los resultados alrededor del valor esperado.

Finalmente, se calcula el CVaR95, definido como el promedio de la tardanza en el subconjunto de escenarios cuyo valor excede el VaR95. Esta métrica captura el desempeño esperado condicionado a eventos extremos y adversos.

Cálculo operacional del CVaR95

El cálculo del CVaR95 se realiza de manera directa a partir de las simulaciones Monte Carlo. En primer lugar, los valores de tardanza total obtenidos en todos los escenarios se ordenan de menor a mayor. A continuación, se identifica el umbral correspondiente al percentil 95, que separa el 5% de escenarios de peor desempeño. Finalmente, se calcula el promedio de la tardanza exclusivamente sobre este subconjunto de escenarios extremos. Este valor constituye el CVaR95 y resume el comportamiento de la solución en situaciones críticas.

Relación con la comparación final de métodos

La evaluación mediante simulación Monte Carlo y CVaR95 constituye el marco común de comparación para todos los métodos analizados. Las soluciones generadas por OR-Tools, ALNS baseline y ALNS guiado por aprendizaje por refuerzo se someten exactamente al mismo procedimiento de simulación, con el mismo número de escenarios y las mismas semillas aleatorias. De este modo, las diferencias observadas en métricas

como la tardanza esperada, la variabilidad y el CVaR95 pueden atribuirse directamente a la calidad estructural de las soluciones y a la capacidad de cada método para producir planes robustos frente a la incertidumbre operacional.

5. Resultados

5.1. Análisis individual por método

El análisis de cada método se realizará en base a las siguientes figuras:

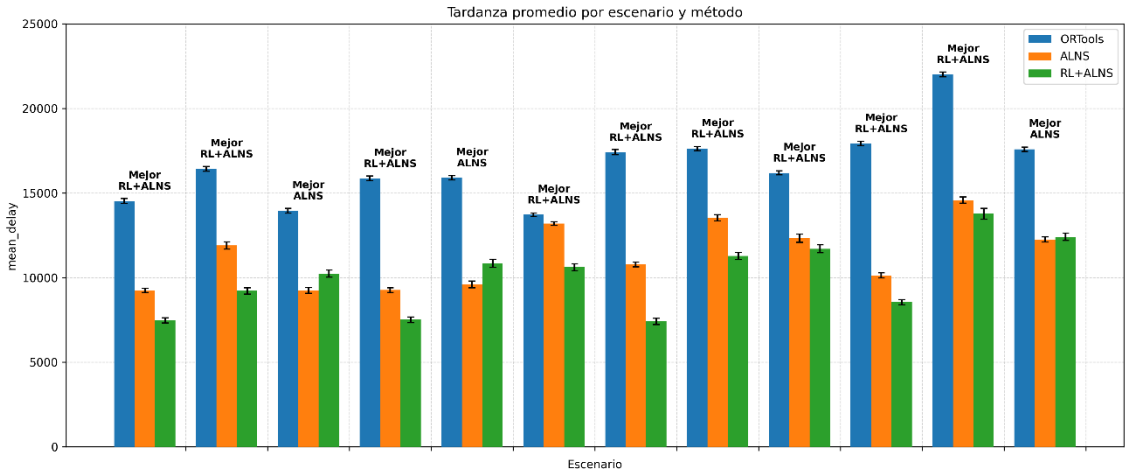

Figura 1: Tardanza promedio por escenario y método. Fuente: Elaboración propia.

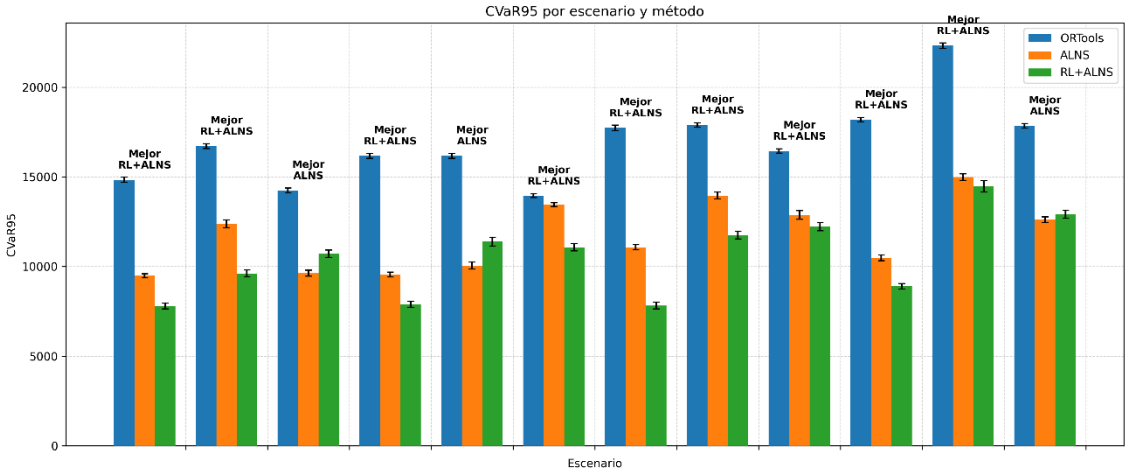

Figura 2: CVaR95 por escenario y método. Fuente: Elaboración propia.

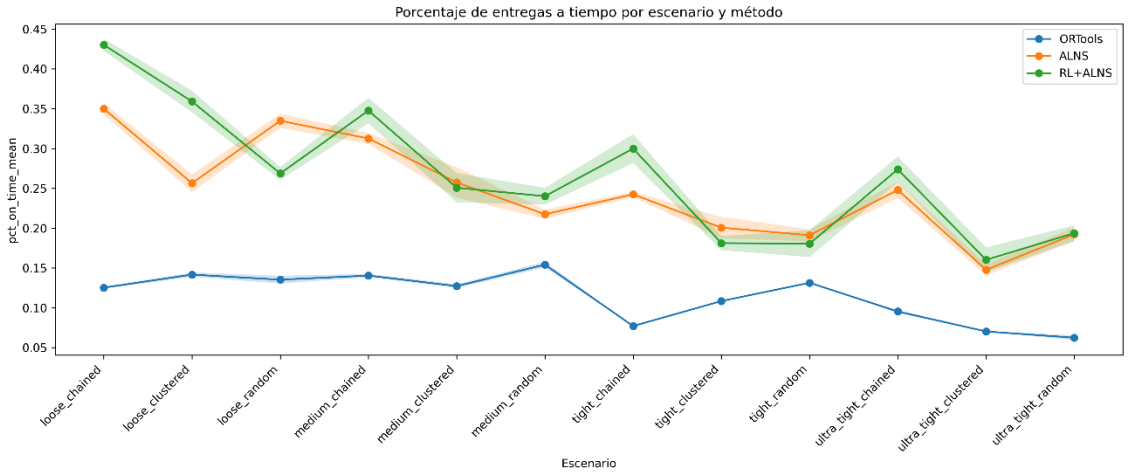

Figura 3: Porcentaje de entregas a tiempo por escenario y método. Fuente: Elaboración propia.

Baseline OR-Tools

El método OR-Tools actúa como línea base determinista y exhibe, de manera consistente, el peor desempeño en términos de tardanza promedio (mean_delay) y riesgo extremo (CVaR95) en todos los escenarios evaluados. Como se observa en la Figura 1, las soluciones obtenidas por OR-Tools presentan valores de mean_delay sustancialmente superiores a los métodos basados en metaheurísticas y aprendizaje por refuerzo, con una degradación particularmente marcada a medida que los escenarios se vuelven más restrictivos (tight y ultra_tight). Esta tendencia se replica de forma casi idéntica en la Figura 2, donde OR-Tools domina sistemáticamente los valores más altos de CVaR95, evidenciando una alta exposición a escenarios de cola bajo incertidumbre.

Desde el punto de vista de la estabilidad, OR-Tools muestra desviaciones estándar (std_delay) relativamente bajas en comparación con los otros métodos. Sin embargo, esta aparente estabilidad es engañosa, ya que está asociada a soluciones

consistentemente malas y no a una buena gestión del riesgo. En otras palabras, el método es estable alrededor de un desempeño deficiente.

En cuanto al nivel de servicio, la Figura 3 revela que OR-Tools obtiene los porcentajes más bajos de entregas a tiempo en todos los escenarios, con valores que decrecen drásticamente en escenarios tight y ultra_tight, llegando a niveles inferiores al 10% en múltiples configuraciones. Esto confirma que el enfoque determinista no logra adaptarse adecuadamente a la incertidumbre incorporada en la evaluación Monte Carlo.

Finalmente, respecto al tiempo computacional, OR-Tools presenta una ventaja clara: el tiempo de resolución es fijo y reducido (60 segundos), independientemente del escenario. No obstante, estos bajos tiempos de cómputo se obtienen a costa de una calidad de solución claramente inferior, lo que limita severamente su aplicabilidad en contextos donde el riesgo y el nivel de servicio son críticos.

Baseline ALNS

El método ALNS muestra una mejora sustancial respecto a OR-Tools en todas las métricas de desempeño. En la Figura 1, ALNS reduce de forma consistente la mean_delay en todos los escenarios, situándose en un nivel intermedio entre OR-Tools y PPO-ALNS. Esta mejora se vuelve especialmente relevante en escenarios loose y medium, donde ALNS logra reducciones significativas de tardanza promedio.

En términos de riesgo, la Figura 2 indica que ALNS también reduce el CVaR95 de manera sistemática frente a OR-Tools, aunque sin alcanzar los niveles más bajos obtenidos por PPO-ALNS en la mayoría de los escenarios. Esto sugiere que ALNS es

capaz de generar soluciones más robustas, pero aún presenta una mayor exposición a eventos extremos que el enfoque híbrido con aprendizaje por refuerzo.

Respecto a la estabilidad ALNS presenta valores de std_delay moderados. En algunos escenarios, particularmente bajo configuraciones clustered o random, la desviación estándar aumenta, reflejando una mayor variabilidad entre realizaciones Monte Carlo. Aun así, esta variabilidad se mantiene dentro de rangos razonables y claramente inferiores a la degradación observada en OR-Tools.

El nivel de servicio obtenido por ALNS, mostrado en la Figura 3, es considerablemente superior al del baseline determinista. En escenarios loose y medium, los porcentajes de entregas a tiempo se sitúan frecuentemente entre 25% y 35%, aunque tienden a disminuir en escenarios tight y ultra_tight. En estos casos, ALNS mantiene un desempeño aceptable, pero es superado de forma consistente por PPO-ALNS.

En cuanto al tiempo computacional, ALNS es el método más costoso. Los tiempos de ejecución son significativamente mayores (del orden de varios minutos por instancia), lo que refleja la complejidad del proceso iterativo de destrucción y reparación. Este costo computacional representa una desventaja práctica, especialmente en aplicaciones donde se requiere una respuesta rápida.

Agente RL (PPO) + ALNS

El método PPO-ALNS presenta el mejor desempeño global en la mayoría de los escenarios y métricas evaluadas. En la Figura 1, se observa que PPO-ALNS alcanza sistemáticamente los menores valores de mean_delay, destacando especialmente en escenarios tight y ultra_tight, donde la brecha respecto a ALNS y OR-Tools se amplía de

manera significativa. Esto evidencia la capacidad del agente de aprendizaje por refuerzo para seleccionar operadores que priorizan soluciones con menor tardanza esperada bajo incertidumbre.

De forma coherente, la Figura 2 muestra que PPO-ALNS obtiene los menores valores de CVaR95 en la gran mayoría de los escenarios, lo que indica una reducción efectiva del riesgo en la cola de la distribución de tardanza. Este resultado es particularmente relevante desde una perspectiva de optimización robusta, ya que demuestra que el método no solo mejora el valor esperado, sino también el comportamiento en los peores escenarios.

En términos de estabilidad, PPO-ALNS presenta valores de std_delay ligeramente superiores a ALNS en algunos escenarios, reflejando una mayor exploración del espacio de soluciones. No obstante, esta mayor variabilidad está asociada a soluciones con mejor desempeño promedio y menor riesgo extremo, por lo que no representa una debilidad estructural del método.

El nivel de servicio, ilustrado en la Figura 3, es donde PPO-ALNS muestra una de sus ventajas más claras. En prácticamente todos los escenarios, este método alcanza los mayores porcentajes de entregas a tiempo, superando de forma consistente a ALNS y duplicando o triplicando los valores obtenidos por OR-Tools en escenarios restrictivos. Este resultado confirma que la política aprendida logra internalizar de manera efectiva el compromiso entre tardanza y cumplimiento de ventanas de tiempo.

Finalmente, en lo relativo al tiempo computacional, PPO-ALNS presenta una ventaja decisiva frente a ALNS. Los tiempos de ejecución son significativamente menores (del orden de decenas de segundos), lo que demuestra que el costo adicional del entrenamiento del agente se ve compensado por una ejecución mucho más eficiente en fase de inferencia, manteniendo al mismo tiempo una calidad de solución superior.

Síntesis

En conjunto, los resultados muestran que OR-Tools es adecuado únicamente como baseline rápido, ALNS ofrece soluciones de buena calidad a un alto costo computacional, y PPO-ALNS logra el mejor equilibrio entre calidad, riesgo, nivel de servicio y tiempo de cómputo, consolidándose como la alternativa más robusta para el problema estudiado bajo incertidumbre.

5.2. Análisis comparativo entre métodos

El análisis conjunto de los resultados evidencia un trade-off claro entre calidad de solución, robustez frente a escenarios adversos y tiempo computacional. Este compromiso se observa de manera consistente al contrastar el desempeño de OR-Tools, ALNS y PPO-ALNS bajo distintos niveles de severidad de los escenarios.
Desde la perspectiva del riesgo, medido a través de CVaR95, la Figura 1 y Figura 2 muestran una separación estructural entre los tres enfoques. OR-Tools presenta sistemáticamente los valores más altos de CVaR95 en todos los escenarios, con una degradación particularmente severa en escenarios tight y ultra_tight, donde los valores de cola superan ampliamente a los métodos basados en ALNS. Esto indica que, bajo

incertidumbre, las soluciones deterministas obtenidas por OR-Tools son altamente vulnerables a realizaciones adversas, lo que limita su aplicabilidad en contextos donde el control del riesgo es un objetivo central.

ALNS clásico logra una reducción significativa del CVaR95 respecto a OR-Tools, posicionándose como una alternativa robusta desde el punto de vista de calidad de solución. En escenarios loose y medium, ALNS alcanza valores de CVaR95 relativamente cercanos a los obtenidos por PPO-ALNS, lo que sugiere que el esquema de destrucción y reparación, junto con el enfriamiento simulado, es capaz de capturar buena parte de la estructura del problema. Sin embargo, en escenarios tight y ultra_tight, la Figura 1 muestra que la brecha entre ALNS y PPO-ALNS se amplía, evidenciando que el enfoque clásico pierde efectividad cuando las ventanas de tiempo se vuelven más restrictivas y la incertidumbre tiene mayor impacto en la cola de la distribución. La dominancia de PPO-ALNS se vuelve especialmente clara en los escenarios más severos. En prácticamente todas las configuraciones tight y ultra_tight, PPO-ALNS obtiene los menores valores de CVaR95, como se observa de forma consistente en la Figura 1 lo que indica una mejor gestión del riesgo extremo. Esta dominancia no es marginal, sino estructural: en varios escenarios ultra_tight, PPO-ALNS reduce el CVaR95 en miles de unidades respecto a ALNS y en más de un orden de magnitud respecto a OR-Tools. Este resultado sugiere que la política aprendida internaliza de manera efectiva la relación entre decisiones locales de ruteo y su impacto en la cola de tardanza, algo que el ALNS clásico no logra capturar de forma adaptativa.

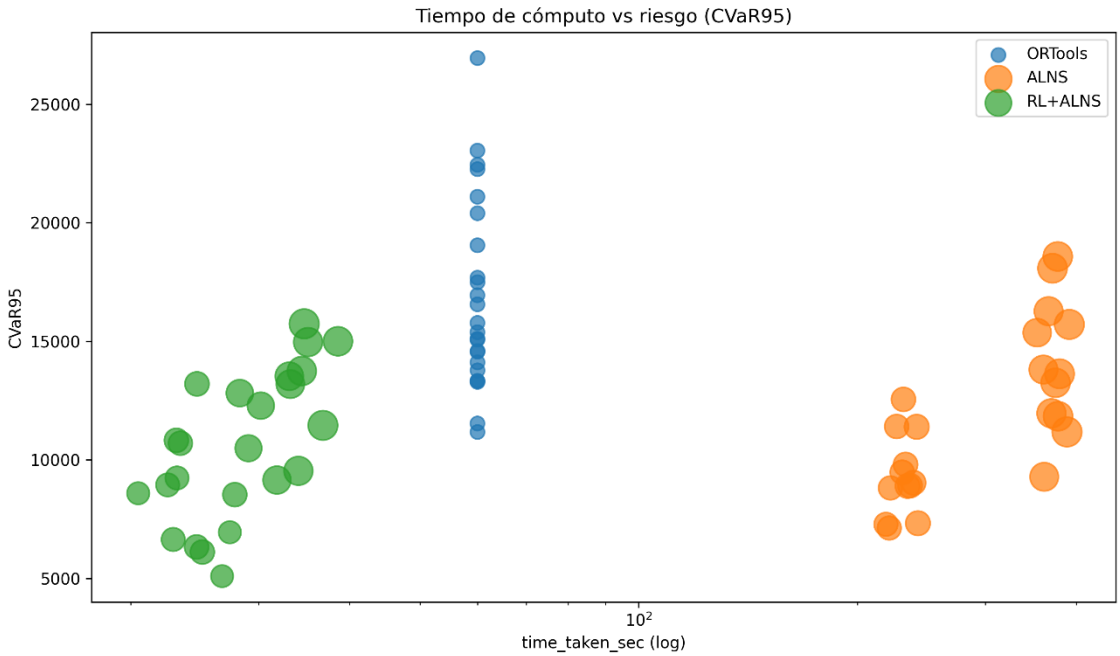

Figura 4:Tiempo de cómputo vs riesgo (CVaR095). Fuente: Elaboración propia.

El trade-off calidad versus tiempo computacional se ilustra claramente en la Figura 4, donde se relaciona el CVaR95 con el tiempo de cómputo en escala logarítmica. OR-Tools ocupa una región de bajo tiempo de cómputo pero alto riesgo, mostrando que su rapidez se obtiene a costa de soluciones frágiles bajo incertidumbre. ALNS, en contraste, se desplaza hacia una región de menor CVaR95, pero con tiempos de cómputo significativamente mayores, del orden de cientos de segundos. PPO-ALNS logra un posicionamiento claramente favorable, combinando valores de CVaR95 bajos con tiempos de cómputo cercanos a los de OR-Tools y muy inferiores a los de ALNS, lo que refleja una frontera de Pareto más eficiente.

Este resultado se ve reforzado por la Figura 5, que muestra la distribución de ticks computacionales. ALNS presenta una dispersión amplia y valores extremadamente altos

de ticks, lo que evidencia un costo computacional elevado y variable. PPO-ALNS, en cambio, exhibe una distribución mucho más concentrada y varios órdenes de magnitud inferior, lo que confirma que la selección guiada de operadores reduce significativamente el esfuerzo computacional sin sacrificar calidad de solución.

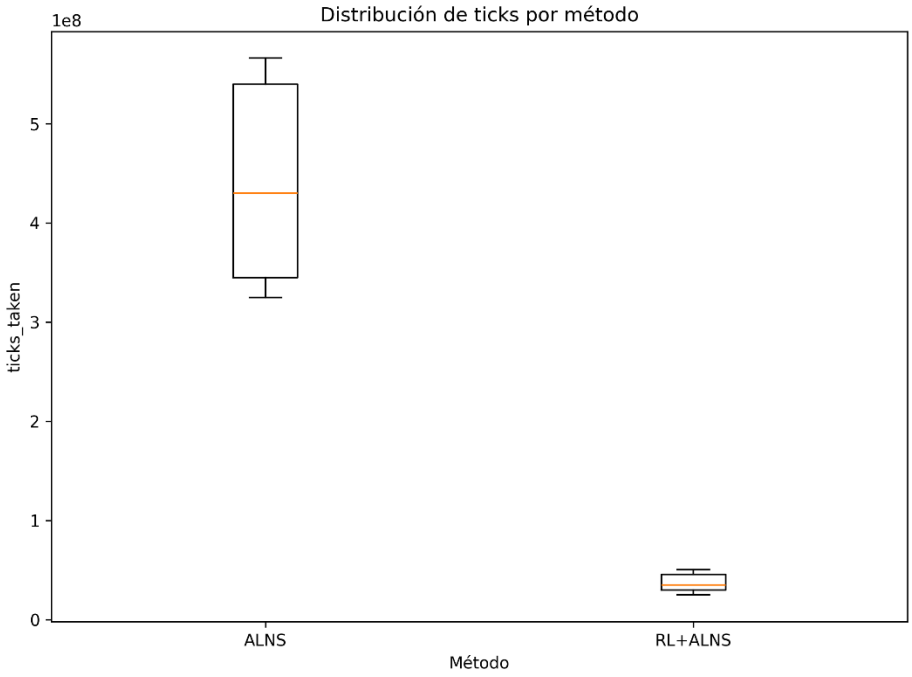

Figura 5: Distribución de ticks por método. Fuente: Elaboración propia.

Las limitaciones estructurales de OR-Tools quedan de manifiesto no solo en términos de CVaR95, sino también en su incapacidad para adaptarse a escenarios severos. Al tratarse de un enfoque determinista, las soluciones generadas no incorporan explícitamente la variabilidad estocástica evaluada posteriormente mediante simulación Monte Carlo. Como consecuencia, el método es incapaz de anticipar o mitigar eventos extremos, lo que se traduce en colas de tardanza pronunciadas y bajos niveles de servicio, tal como se

observa indirectamente al relacionar CVaR95 con el porcentaje de entregas a tiempo en la Figura 3.

En contraste, ALNS clásico muestra una robustez razonable frente a la incertidumbre, especialmente en escenarios de severidad baja y media. Sin embargo, su carácter no adaptativo limita su capacidad para priorizar operadores de forma diferenciada según el estado de la solución y el contexto del escenario. PPO-ALNS supera esta limitación al aprender una política que ajusta dinámicamente la selección de operadores, lo que se traduce en una mejor robustez global, particularmente en términos de CVaR95, sin incurrir en el elevado costo computacional del ALNS clásico.

En síntesis, el análisis comparativo confirma que PPO-ALNS define una frontera superior en el trade-off entre calidad de solución, control del riesgo extremo y tiempo de cómputo. ALNS clásico constituye una alternativa robusta pero costosa, mientras que OR-Tools, pese a su eficiencia computacional, presenta limitaciones estructurales severas bajo incertidumbre, quedando relegado a un rol de baseline determinista.

5.3. Análisis de comportamiento de operadores

El análisis del uso de operadores revela diferencias estructurales profundas entre el ALNS clásico y el enfoque PPO_ALNS. En el caso del ALNS baseline, la selección de operadores responde a un esquema esencialmente balanceado y cuasi-aleatorio, producto de pesos iniciales uniformes y mecanismos de actualización poco discriminativos. Esto se observa claramente en la Figura 6 de uso acumulado de operadores de reparación y destrucción para ALNS, donde las proporciones de uso se mantienen cercanas entre sí a

lo largo de toda la ejecución, sin que emerja una preferencia marcada por un operador específico. En particular, los operadores de reparación greedy, time windows y regret-2 presentan participaciones similares, al igual que los operadores de destrucción random 10%, random 20%, random 40%, relocate y worst.

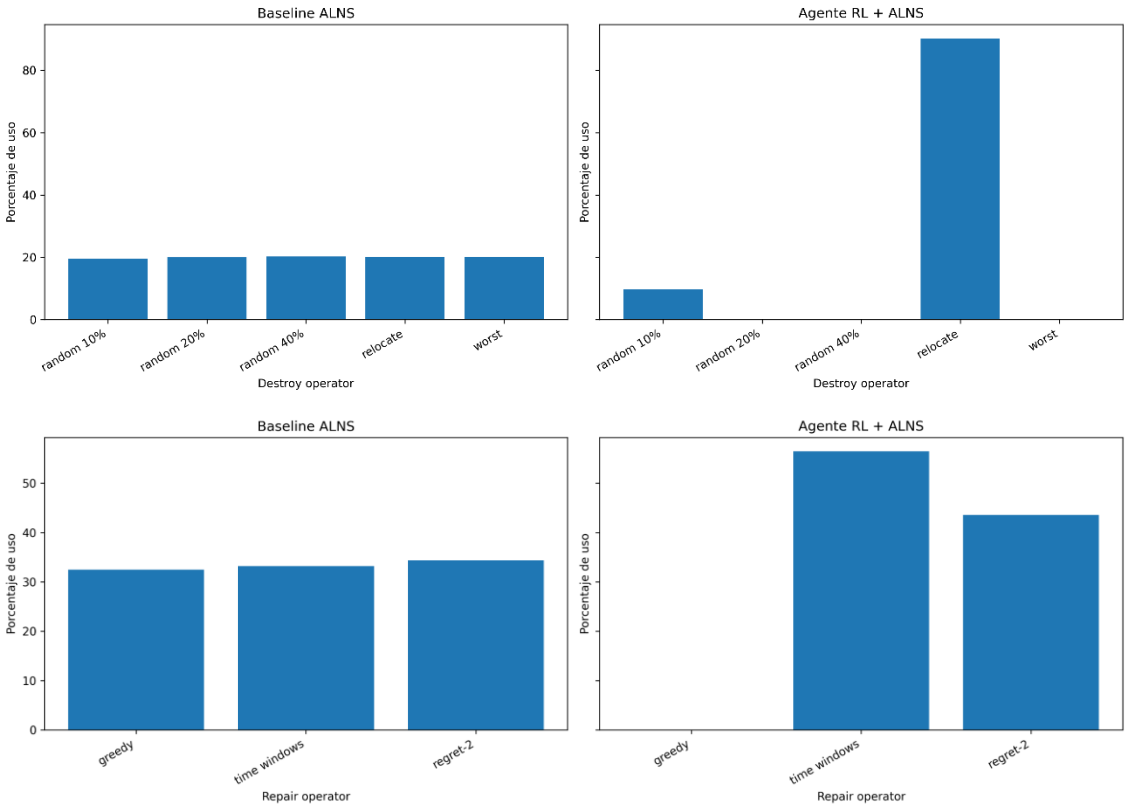

Figura 6: Comparación de frecuencia de uso de operadores ALNS según método. Fuente: Elaboración propia.

En contraste, el comportamiento del agente PPO_ALNS muestra una política de selección altamente estructurada y no uniforme. Los gráficos de uso acumulado evidencian una fuerte concentración en un subconjunto reducido de operadores, tanto de reparación como de destrucción. En el caso de los operadores de reparación, el agente prioriza de manera dominante el operador time windows, seguido en menor medida por

regret-2, mientras que el operador greedy es prácticamente descartado. Esta selección sugiere que el agente ha aprendido que, bajo incertidumbre y ventanas de tiempo estrictas, la reparación guiada explícitamente por restricciones temporales resulta más efectiva para reducir el riesgo extremo medido por CVaR95.

Un patrón aún más marcado se observa en los operadores de destrucción. Mientras que el ALNS baseline distribuye el uso de manera relativamente uniforme entre destrucciones aleatorias y estructurales, el agente PPO_ALNS concentra casi todo su uso en el operador relocate, con una participación marginal del operador random 10% y una exclusión casi total de random 20%, random 40% y worst. Este comportamiento indica una especialización clara hacia destrucciones controladas y localizadas, coherentes con escenarios tight y ultra_tight, donde perturbaciones excesivas tienden a degradar la factibilidad temporal y aumentar la cola de tardanza.

La especialización de operadores se vuelve particularmente relevante al considerar escenarios severos. En contextos tight y ultra_tight, la combinación dominante time windows + relocate resulta consistente con la necesidad de ajustes finos y dirigidos sobre rutas ya cercanas a la factibilidad. La política aprendida evita destrucciones agresivas y operadores de reparación miopes, privilegiando acciones que preservan estructura mientras corrigen violaciones críticas de ventanas de tiempo. Esta especialización no emerge en el ALNS clásico, donde la ausencia de una señal de aprendizaje global impide diferenciar entre escenarios de distinta severidad.

La evidencia de adaptación temporal en la Figura 7 se aprecia con claridad en los gráficos de evolución por iteración de operadores. En el ALNS baseline, la selección de

operadores a lo largo de las iteraciones muestra una dispersión homogénea, sin cambios estructurales en el tiempo. Los puntos asociados a distintos operadores se mantienen distribuidos a lo largo de todo el horizonte de iteraciones, reflejando una exploración persistente pero no dirigida.

Por el contrario, en PPO_ALNS la evolución temporal revela fases claramente diferenciadas. En etapas tempranas, el agente utiliza una combinación más amplia de operadores, consistente con un período de exploración. Sin embargo, a medida que avanzan las iteraciones, la política converge rápidamente hacia un subconjunto estable de operadores, principalmente relocate en destrucción y time windows en reparación. Esta convergencia temporal constituye evidencia directa de adaptación dinámica, donde el agente ajusta su comportamiento en función del estado de la solución y del retorno esperado en términos de reducción de CVaR95.

En conjunto, los resultados muestran que el ALNS clásico opera bajo un paradigma de robustez genérica, basada en diversidad de operadores, pero sin especialización explícita. PPO_ALNS, en cambio, aprende una política selectiva y adaptativa, capaz de identificar qué operadores son más efectivos bajo condiciones de alta restricción temporal y riesgo. Esta capacidad de especialización y adaptación temporal explica en gran medida la superioridad observada de PPO_ALNS en escenarios tight y ultra_tight, tanto en términos de CVaR95 como de eficiencia computacional.

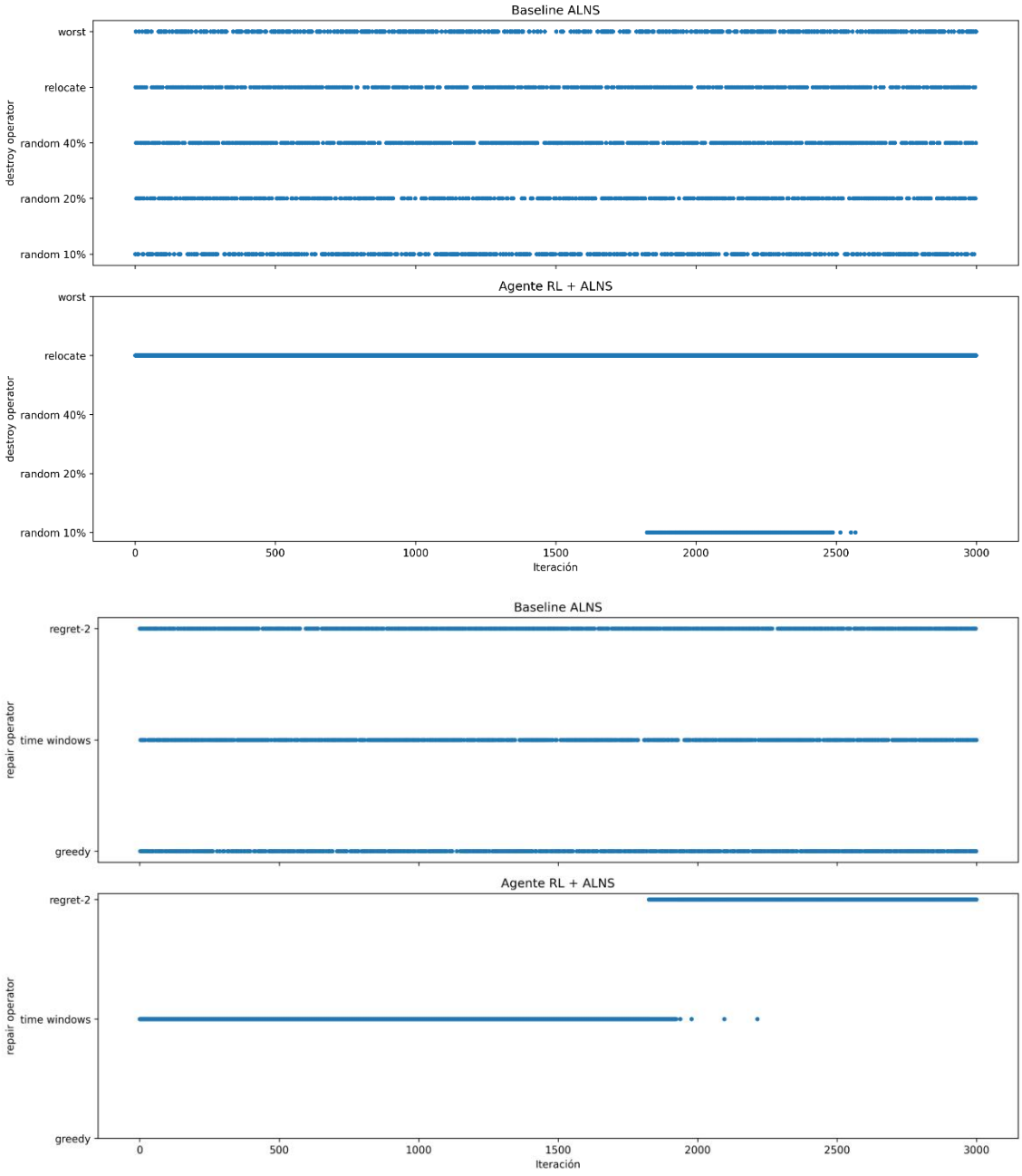

Figura 7: Utilización de operadores en el tiempo. Fuente: Elaboración propia.

5.4. Análisis de ticks y eficiencia computacional

El análisis de los ticks acumulados revela una diferencia estructural muy marcada en el esfuerzo computacional requerido por ALNS y PPO_ALNS. En la Figura 8, el ALNS clásico exhibe un crecimiento prácticamente lineal y continuo a lo largo de todas las iteraciones, alcanzando valores del orden de 3×10^8 a 5×10^8 ticks al final de la ejecución. Este comportamiento se mantiene tanto para el día 8 como para el día 9, con una dispersión relativamente acotada entre corridas, lo que indica que el método consume recursos de manera constante e intensiva durante todo el horizonte de búsqueda.

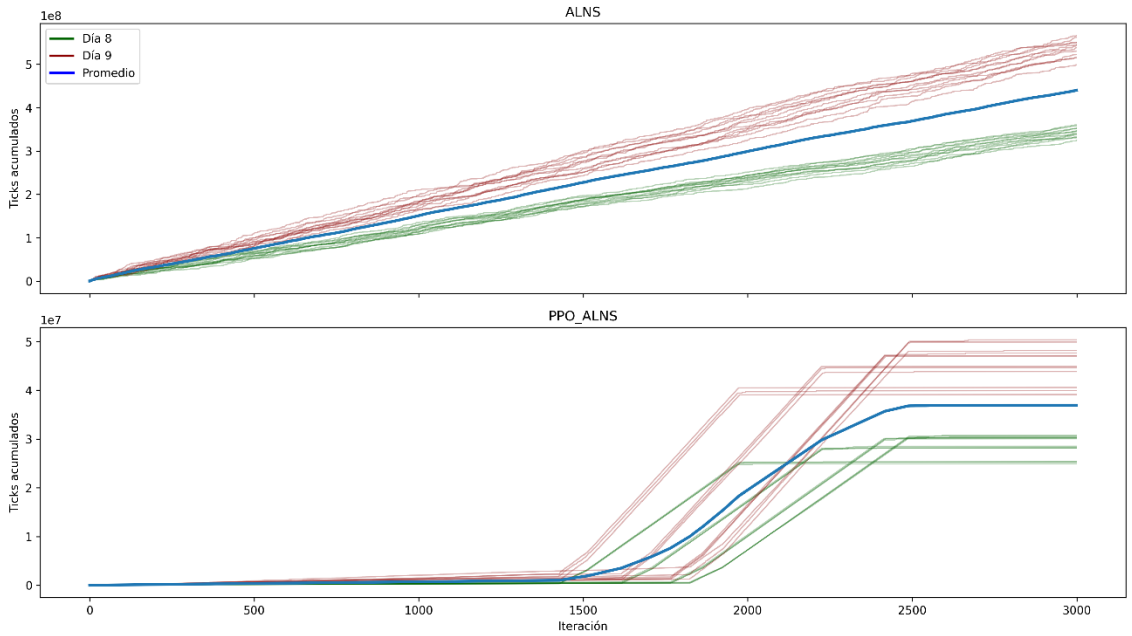

Figura 8: Ticks acumulados vs número de iteración por método. Fuente: Elaboración propia.

En contraste, PPO_ALNS presenta un patrón radicalmente distinto. En la figura temporal de ticks acumulados, se observa una fase inicial prolongada con un crecimiento casi nulo del contador de ticks, seguida de una transición relativamente abrupta donde el

esfuerzo computacional aumenta de forma concentrada, para luego estabilizarse nuevamente. El valor final de ticks es uno o incluso dos órdenes de magnitud menor que el observado en ALNS, situándose en el rango de 2.5×10^7 a 5×10^7 ticks. Este patrón escalonado refleja que PPO_ALNS no explora de forma exhaustiva el espacio de soluciones en cada iteración, sino que activa fases intensivas de cómputo solo cuando la política aprendida identifica oportunidades claras de mejora.

La relación entre ticks y mejora en CVaR95 resulta clave para interpretar la eficiencia relativa de ambos métodos. En ALNS, el incremento sostenido de ticks no se traduce en mejoras proporcionales del CVaR95, especialmente en escenarios tight y ultra_tight. Tal como se observa al contrastar las figuras de ticks con las de CVaR95, gran parte del esfuerzo computacional adicional se invierte en explorar vecindarios que producen mejoras marginales o incluso neutras desde la perspectiva del riesgo extremo. Esto sugiere rendimientos decrecientes claros: cada unidad adicional de esfuerzo computacional aporta cada vez menos reducción en la cola de la distribución de tardanza.

Por el contrario, en PPO_ALNS la mayor parte de la reducción de CVaR95 se logra con una fracción muy reducida de ticks totales. La fase de crecimiento acelerado de ticks coincide temporalmente con el período en que se observan las mayores mejoras en CVaR95, tras lo cual el método converge rápidamente y deja de consumir recursos de forma intensiva. Esto indica una relación mucho más eficiente entre esfuerzo computacional y reducción de riesgo, donde los ticks se concentran en iteraciones de alto impacto y no se desperdician en exploración poco informativa.

Desde un punto de vista práctico y operacional, estas diferencias tienen implicancias directas. El ALNS clásico, si bien es capaz de producir soluciones de buena calidad, requiere un presupuesto computacional elevado y sostenido, lo que dificulta su uso en contextos donde el tiempo de respuesta o el costo computacional son restricciones relevantes. Además, la falta de una señal clara de convergencia implica que resulta complejo definir criterios operacionales de detención sin incurrir en pérdidas significativas de calidad.

En cambio, PPO_ALNS ofrece un perfil computacional mucho más atractivo para uso operacional. La reducción drástica en ticks totales, junto con la concentración del esfuerzo en fases específicas, permite obtener soluciones con bajo CVaR95 en tiempos significativamente menores y con mayor previsibilidad del costo computacional. Esto habilita su aplicación en escenarios de planificación recurrente o casi en tiempo real, donde es necesario balancear explícitamente calidad de solución y consumo de recursos.

En síntesis, el análisis de ticks acumulados y temporales confirma que PPO_ALNS no solo domina a ALNS en términos de CVaR95, sino que lo hace con una eficiencia computacional muy superior. La relación ticks/mejora de CVaR95 es claramente favorable al enfoque con aprendizaje por refuerzo, lo que refuerza su idoneidad para aplicaciones operacionales bajo incertidumbre y restricciones de tiempo.

6. Conclusiones

El objetivo principal de este trabajo fue evaluar y comparar distintos enfoques de resolución para un problema de ruteo con ventanas de tiempo bajo incertidumbre, incorporando explícitamente métricas de riesgo mediante simulación Monte Carlo y CVaR95. Los resultados obtenidos permiten extraer varias conclusiones relevantes tanto desde una perspectiva metodológica como aplicada.

En primer lugar, el uso de CVaR95 como métrica principal de comparación resultó clave para diferenciar de manera clara el desempeño de los métodos. A diferencia de métricas basadas únicamente en valores esperados, CVaR95 permitió capturar el comportamiento de cola de la distribución de tardanza, revelando diferencias sustanciales entre enfoques que no siempre son evidentes al observar solo la tardanza promedio. En este sentido, los resultados confirman que optimizar soluciones bajo incertidumbre requiere métricas explícitamente sensibles al riesgo extremo.

El modelo OR-Tools, utilizado como baseline determinista, mostró un desempeño consistentemente inferior en todos los escenarios evaluados. Si bien presenta tiempos de cómputo bajos y estables, sus valores de CVaR95 y tardanza promedio son sistemáticamente los más altos, especialmente en escenarios tight y ultra_tight. Esto evidencia una limitación estructural del enfoque determinista: al no incorporar la variabilidad estocástica durante la construcción de la solución, las rutas generadas resultan altamente vulnerables a realizaciones adversas. En consecuencia, OR-Tools queda restringido a un rol de referencia base, pero no constituye una alternativa adecuada cuando el control del riesgo y el nivel de servicio son objetivos prioritarios.

El ALNS clásico representa una mejora significativa respecto al baseline determinista. Los resultados muestran reducciones sustanciales tanto en tardanza promedio como en CVaR95, junto con un aumento claro en el porcentaje de entregas a tiempo. Esto confirma la efectividad del esquema de destrucción y reparación para explorar soluciones de mayor calidad y robustez. Sin embargo, el análisis también revela que esta mejora se obtiene a un costo computacional muy elevado. El crecimiento casi lineal y sostenido de los ticks, junto con tiempos de ejecución del orden de cientos de segundos, indica que gran parte del esfuerzo computacional se destina a exploración con rendimientos decrecientes, especialmente en escenarios severos.

El enfoque PPO_ALNS emerge como el método con mejor desempeño global. En la mayoría de los escenarios, y de forma especialmente marcada en los escenarios tight y ultra_tight, PPO_ALNS alcanza los menores valores de CVaR95, reduciendo de manera significativa el riesgo extremo frente a ALNS y OR-Tools. Este mejor desempeño se acompaña además de mayores niveles de servicio y tardanzas promedio más bajas.

Desde el punto de vista computacional, PPO_ALNS logra estos resultados con una fracción del esfuerzo requerido por ALNS, evidenciado por la reducción de uno a dos órdenes de magnitud en ticks acumulados y tiempos de cómputo.

El análisis del comportamiento de operadores explica en gran medida esta superioridad. Mientras el ALNS clásico mantiene una selección casi uniforme de operadores, el agente PPO_ALNS aprende una política altamente especializada y adaptativa. La concentración en operadores como time windows para reparación y relocate para destrucción, junto con la exclusión de operadores menos efectivos en escenarios restrictivos, permite focalizar

el esfuerzo computacional en acciones de alto impacto sobre CVaR95. La evolución temporal de los operadores muestra además una clara fase de adaptación y convergencia, lo que evidencia aprendizaje efectivo y no meramente heurístico.

En conjunto, los resultados permiten concluir que la integración de aprendizaje por refuerzo en el marco de ALNS no solo mejora la calidad de las soluciones en términos de riesgo y nivel de servicio, sino que redefine la frontera de eficiencia entre calidad y esfuerzo computacional. PPO_ALNS ofrece un equilibrio claramente superior, lo que lo convierte en una alternativa particularmente atractiva para aplicaciones operacionales bajo incertidumbre, donde el tiempo de respuesta y el control del riesgo son restricciones críticas.

Limitaciones del estudio

A pesar de los resultados positivos, este trabajo presenta una serie de limitaciones que deben ser consideradas al interpretar los resultados y al proyectar futuras extensiones.

En primer lugar, el entrenamiento del agente PPO se realizó sobre un conjunto acotado de días y escenarios. Si bien los resultados muestran buena generalización entre los días evaluados, no se puede garantizar que la política aprendida mantenga el mismo nivel de desempeño frente a distribuciones de demanda o patrones de incertidumbre significativamente distintos a los utilizados en el entrenamiento.

En segundo lugar, la función de recompensa utilizada por el agente prioriza principalmente métricas relacionadas con tardanza y riesgo. Otros objetivos relevantes en problemas reales de ruteo, como costos operacionales directos, balance de carga entre vehículos o equidad entre clientes, no fueron incorporado. Esto puede limitar la

aplicabilidad directa del enfoque en contextos donde múltiples objetivos compiten de manera explícita.

Otra limitación importante es el costo asociado al entrenamiento del agente de aprendizaje por refuerzo. Si bien la fase de inferencia es altamente eficiente, el proceso de entrenamiento requiere una inversión computacional significativa y un diseño cuidadoso del entorno, la representación del estado y la función de recompensa. Este costo inicial puede no ser justificable en todos los contextos operacionales.

Adicionalmente, el conjunto de operadores considerado en el ALNS es limitado y fue definido a priori. Aunque el agente aprende a seleccionar operadores de manera efectiva, no se explora la generación automática de nuevos operadores ni la adaptación dinámica del conjunto de vecindarios, lo que podría ofrecer mejoras adicionales.

Finalmente, la evaluación bajo incertidumbre se basa en un esquema específico de simulación Monte Carlo y supuestos particulares sobre las distribuciones de los tiempos. Cambios en estos supuestos podrían afectar los valores absolutos de CVaR95 y, potencialmente, la jerarquía relativa entre métodos, lo que abre la necesidad de validar el enfoque bajo modelos de incertidumbre alternativos.

En síntesis, el proyecto demuestra que la combinación de ALNS con aprendizaje por refuerzo constituye una estrategia poderosa y eficiente para problemas de ruteo bajo incertidumbre. No obstante, su adopción práctica requiere considerar cuidadosamente las limitaciones señaladas y evaluar su desempeño en contextos más amplios y diversos.

7. Bibliografía

Aravena Cabrera, D. A. (2023). *TOMA DE DECISIONES PARA FLOTAS DE VEHÍCULOS ELÉCTRICOS.* Santiago de Chile.

Cao, Z., Zhu, Z., Lu, W., & Zhang, S. (2025). Risk-Aware Vessel Scheduling and Routing Optimization with CVaR and LSTM-MSNet Prediction. *Journal of Marine Science & Engineering, 2025, Vol 13, Issue 2, p207*.

Lin, B., Ghaddar, B., & Nathwani, J. (2022). Deep Reinforcement Learning for the Electric Vehicle Routing Problem With Time Windows. *IEEE Transactions on Intelligent Transportation Systems, 23(8), 11528–11538*.

OpenAI. (s.f.). *spinningup.openai.com*. Obtenido de spinningup.openai.com: https://spinningup.openai.com/en/latest/algorithms/ppo.html

Wouda, N. (s.f.). *alns.readthedocs.io*. Obtenido de alns.readthedocs.io: https://alns.readthedocs.io/en/latest/examples/alns_features.html

Zhang, Q. (2024). CVaR-Constrained Policy Optimization for Safe Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*.

Zong, Z., Xia, T., Meng, Z., & Li, Y. (2024). Reinforcement Learning for Solving Multiple Vehicle Routing Problem with Time Window. *ACM Transactions on Intelligent Systems and Technology, Volume 15, Issue 2*.

8. Anexos

8.1. Repositorio GitHub con código del proyecto

https://github.com/gabalvarezmc/Capstone_CVRPTW_RL_ALNS

8.2. Tabla de resultados

https://github.com/gabalvarezmc/Capstone_CVRPTW_RL_ALNS/tree/main/results/20260106_025250

8.3. Formulación matemática CVRPTW

https://github.com/gabalvarezmc/Capstone_CVRPTW_RL_ALNS/blob/main/Formulacion_capstone.pdf