Logística bajo Incertidumbre

Metaheurísticas para el ruteo de vehículos con flota heterogénea, ventanas de tiempo flexibles y clientes prioritarios

Diego Armando Soto de La Vega
Tobías Alfonso Parodi Camaño
Jorge Mario López Pereira

Ediciones Transformar

ISBN 978-956-08230-6-9

Ediciones Transformar

Logística bajo Incertidumbre:
Metaheurísticas para el ruteo de vehículos con flota heterogénea,
ventanas de tiempo flexibles y clientes prioritarios
Diego Armando Soto de La Vega, Tobías Alfonso Parodi Camaño y Jorge Mario López Pereira

Una publicación de Ediciones Transformar, sello editorial perteneciente a Centro Transformar SpA/Red Internacional de Investigadores en Educación (REDIIE), Chile.
Sitio web: https://centrotransformar.cl/ediciones-transformar/

Editor general: Fernando Vera, PhD fernandovera@rediie.cl
Diseño y plataformas digitales: Jorge Vargas soporte@rediie.cl

Licencia de uso
© Los/as editores/as (si corresponde) y los/as autores/as 2025. Este libro es una publicación de acceso abierto.

Acceso Abierto: Este libro está licenciado bajo los términos de la Licencia Internacional Creative Commons Chile: Atribución-NoComercial-SinDerivadas 4.0 Internacional BY-NC-ND, que permite compartir el contenido (por ejemplo, publicarlo en una web o redes sociales), sin modificarlo y sin fines comerciales, siempre que menciones al autor.
La editorial, las personas autoras y el editor asumen que los consejos e información contenidos en este libro son considerados veraces y precisos en la fecha de su publicación. Ni la editorial ni los/as autores/as o editores/as ofrecen garantía, expresa o implícita, respecto al material incluido ni por errores u omisiones que puedan haberse producido. La editorial se mantiene neutral respecto a reclamaciones jurisdiccionales en mapas publicados y afiliaciones institucionales.

Las imágenes u otro material de terceros que se incluyen en este libro están cubiertos por la licencia Creative Commons del libro, a menos que se indique lo contrario en la línea de crédito correspondiente. Si algún material no está incluido bajo la licencia Creative Commons del libro y su uso previsto no está permitido por la legislación vigente o excede el uso permitido, será necesario obtener permiso directamente del titular de los derechos de autor. Esta obra, perteneciente a la marca Transformar®, es publicada por Ediciones Transformar, sello editorial de Centro Transformar SpA.

Esta obra, perteneciente a la marca Transformar®, es publicada por Ediciones Transformar, sello editorial de Centro Transformar SpA.
ISBN 978-956-08230-6-9
Ediciones Transformar 978-956-08230
© Todos los derechos reservados.

Biografía de autores

DIEGO ARMANDO SOTO DE LA VEGA
Ingeniero Industrial, Magíster en Ingeniería de Producción y MBA en Gestión Empresarial. Con experiencia en gestión de datos, inteligencia artificial y liderazgo corporativo, cuento con una trayectoria sólida en el uso de los datos como activos estratégicos para las organizaciones. Me especializo en la implementación de soluciones tecnológicas avanzadas —advanced analytics, inteligencia artificial y big data— para resolver de forma efectiva desafíos empresariales, liderando equipos multidisciplinarios y transformando datos en insights valiosos para la toma de decisiones. Tengo experiencia consolidada en consultoría estratégica, logística, cadena de suministro y gestión de producción, y he participado en más de 50 proyectos para empresas brasileñas y multinacionales de sectores tales como; financiero, automotriz, petróleo y combustibles, bebidas, siderurgia, retail y servicios.
Email: die-soto@hotmail.com | ORCID: https://orcid.org/0009-0007-2213-7429

TOBÍAS ALFONSO PARODI CAMAÑO
Ingeniero Industrial, MBA y candidato a Doctor en Innovación, con más de 10 años de experiencia en consultoría, mentoría y coordinación de proyectos de ciencia y tecnología. Docente de tiempo completo de la Universidad de Córdoba. Ha liderado procesos de formación, de estructuración de modelos de negocio y de aceleración e innovación empresarial en programas como Fondo Emprender. Ha dictado cursos de emprendimiento e innovación en universidades como Córdoba, CUN y UNAD, y ha asesorado programas de cámaras de comercio.
Email: tobiasparodic@correo.unicordoba.edu.co |
Orcid: https://orcid.org/0000-0002-1723-5717

JORGE MARIO LÓPEZ PEREIRA
Ingeniero Industrial con Maestría en Ingeniería Industrial y especialización en Informática y Telemática. Profesor Titular de la Universidad de Córdoba, con más de 15 años de experiencia en docencia e investigación. Sus líneas de trabajo se enfocan en la analítica y la ciencia de datos, la investigación de operaciones y la optimización de cadenas de suministro. Es miembro activo del Grupo de Investigación en Aplicación de Teoría de Restricciones (GITOC). Su motivación para escribir este libro nace de la necesidad de cerrar la brecha entre el análisis técnico y la toma de decisiones gerenciales, haciendo accesible la inteligencia predictiva para líderes y estudiantes.
Email: jotalopez@correo.unicordoba.edu.co | ORCID: https://orcid.org/0000-0001-7317-8772

Contenido

Prólogo 8

Resumen 9

Introducción 10

Capítulo 1. El problema hoy: ruteo bajo incertidumbre y servicio diferenciado 12

1.1. Logística contemporánea y presión por nivel de servicio (última milla, tiempos, penalizaciones) 12

1.2. Formulación conceptual del problema base del modelo 13

1.3. Componentes de costo total (fijo, variable, espera, penalizaciones, horas extra) 14

1.4. Preguntas de investigación y contribución (robustez, confiabilidad, replicabilidad) 15

1.5. Estructura del modelo de investigación y orientación metodológica. 16

Capítulo 2. Estado del arte: rutas, incertidumbre, y solución híbrida 17

2.1. De VRP "clásico" a VRP "real": incertidumbre, dinamismo y servicio 17

2.2. Incertidumbre operacional y time-dependent routing: el tráfico ya es parte del modelo 17

2.3. Soft time windows: penalizar para modelar servicio (no solo restringir) 18

2.4 Servicio diferenciado y última milla: presencia incierta, reintentos y priorización 18

2.5. Métodos dominantes en la literatura reciente: ALNS y los operadores como unidad de diseño 19

2.6. Analítica y machine learning en VRP: de la promesa a la evaluación replicable 19

2.7. Tendencias tecnológicas que reconfiguran el ruteo: EVRP y drones 20

2.8. Síntesis de brechas (gap) y posicionamiento del modelo propuesto de VRP 20

Capítulo 3. Modelo actualizado y diseño experimental reproducible 21

3.1. Modelo matemático base del modelo 21

3.2. Incertidumbre operacional: demanda/servicio/viaje como variables aleatorias (supuestos y alternativas) 23

3.3. Función objetivo y extensiones modernas 24

3.4. Instancias y evaluación comparativa: cómo construir bancos de prueba realistas (más allá de ajustes ad-hoc) 24

3.5. Protocolo de evaluación: métricas, comparación estadística, transparencias de cómputo 25

3.6. Buenas prácticas contemporáneas para el ajuste de parámetros (automatización, robustez y reporte reproducible) 26

3.7. Protocolo de ejecución reproducible y transferencia a operación 27

3.7.1. Notas de reporte mínimo para comparación justa 28

3.7.2. El "contrato de ejecución": qué se debe congelar 29

3.7.3. Congelamientos para horizonte rodante y reoptimización 29

3.7.4. Contrato de ejecución: comparabilidad y replicabilidad como condiciones de credibilidad 30

3.7.5. De la instancia “de papel” a los datos operativos 30

3.7.6. Operación dinámica: estabilidad, calidad de servicio y “costo del cambio” 30

3.7.7. Gemelos digitales como puente: evaluación continua y transferencia 31

3.7.8. Transparencia experimental: registro, versionado y trazabilidad 31

Capítulo 4. Métodos de solución: de metaheurísticas clásicas a enfoques híbridos y optimización asistida por aprendizaje 32

4.1. Línea base: heurísticas constructivas y prioridades por ventanas 32

4.1.1. Heurísticas constructivas clásicas (adaptadas a ventanas) 32

4.1.2. Prioridades por ventanas: EDF y variantes como políticas 33

4.1.3. Plantilla mínima reproducible para un método de referencia (pseudocódigo) 33

4.2. Motores metaheurísticos comparables 34

4.2.1. Métodos de trayectoria: Simulated Annealing (SA) y Tabu Search (TS) 34

4.2.2. Métodos poblacionales: Genetic Algorithms (GA) 35

4.2.3. Enjambres: Particle Swarm Optimization (PSO) y adaptaciones discretas 36

4.2.4. Bio-inspiradas y el enfoque “Ébola” como antecedente histórico del caso 36

4.3. Marco ALNS moderno (destroy/repair, pesos adaptativos, aceptación, intensificación/diversificación) 37

4.3.1. Componentes mínimos del ALNS 37

4.3.2. Diseño de operadores para ventanas y holguras 38

4.3.3. Plantilla operativa (pseudo-código) de ALNS 38

4.4. Híbridos y metaheurísticas ligeras (cuando conviene mezclar MIP con heurística) 38

4.4.1. Patrones útiles de matheurística ligera 39

4.4.2. Criterios prácticos para decidir “cuándo conviene” 39

4.5. “Learning-augmented optimization”: cómo IA mejora selección de operadores / inserciones / policies 39

4.5.1. Niveles de integración ML → Solver 39

4.5.2. Tendencias: GNN/RL y aprendizaje de heurísticas 39

4.5.3. Evaluación justa en ML-VRP (prácticas mínimas) 39

4.6. Arquitectura operacional: ajuste dinámico de rutas en tiempo real + integración con un Digital Twin 40

4.6.1. Ciclo operativo de ajuste dinámico de rutas 40

4.6.2. Componentes mínimos de un Digital Twin para ruteo 40

Capítulo 5. Resultados, implicaciones prácticas y agenda de investigación 41

5.1. Estructura de reporte de resultados (calidad, tiempo, robustez, servicio a prioritarios) 41

5.1.1. Bloque A: Calidad y estructura de la solución 41

5.1.2. Bloque B: Nivel de servicio y priorización 42

5.1.3. Bloque C: Tiempo computacional y presupuesto de cómputo 42

5.1.4. Bloque D: Robustez y confiabilidad bajo escenarios 43

5.2. Lectura gerencial: compensaciones costo–servicio–confiabilidad (qué decisiones cambian el resultado) 44

5.2.1. Compensación 1: costo vs nivel de servicio 45

5.2.2. Compensación 2: eficiencia vs robustez (variabilidad) 45

5.2.3. Compensación 3: planificación offline vs ajuste dinámico de rutas online 45

5.2.4. Qué decisiones 'mueven la aguja' (resumen ejecutivo) 45

5.3. Guía de implementación en empresa (datos necesarios, telemetría, reglas de negocio, SLAs) 46

5.3.1. Datos mínimos (sin los cuales el modelo falla) 46

5.3.2. Telemetría y eventos (para operación 2025+) 46

5.3.3. Reglas de negocio típicas y parametrización 46

5.3.4. Integración técnica (mínimo viable) 46

5.4. Limitaciones y próximos pasos (multi-producto, sostenibilidad, automatización, equidad de servicio) 46

5.4.1. Limitaciones típicas 47

5.4.2. Agenda de investigación emergente (líneas priorizadas) 47

Glosario (términos operativos y de optimización) 47

Lista de siglas 49

Referencias 49

Prólogo

La logística moderna ha trascendido su concepción original como un mero problema estático de planificación. En la actualidad, el proceso de enrutamiento se desarrolla en un ecosistema dinámico caracterizado por la congestión vehicular, una profunda incertidumbre, la presión de promesas de servicio cada vez más estrictas, la afluencia de datos en tiempo real y la imperiosa necesidad de sostenibilidad. Dentro de este complejo escenario, el problema de enrutamiento de vehículos (VRP) ya no puede reducirse a un simple ejercicio de optimización matemática; se erige en un sofisticado sistema de toma de decisiones que interactúa intrínsecamente con la operación, la tecnología y la gestión empresarial.

Este libro, resultado de investigación, surge precisamente de la tensión inherente a esta realidad: por un lado, la exigencia de mantener un fundamento matemático riguroso y verificable; por otro, la urgencia de proponer metodologías implementables y auditables en condiciones operativas reales. En este sentido, el libro constituye una evolución y ampliación del trabajo de grado “Metaheurísticas aplicadas al Problema Dinámico de Ruteo de Vehículos con Demandas y Tiempos Estocásticos, Flota Heterogénea, Ventanas de Tiempo Flexibles y Clientes Preferenciales (SFMVRPMSTWTDCP)”, integrando sus aportes dentro de un marco actualizado de investigación y aplicación.

El texto se estructura según una lógica de actualización responsable. Su objetivo no es suplantar el conocimiento clásico del VRP, sino reorganizarlo y ampliarlo mediante la incorporación de criterios de rigor contemporáneos: comparabilidad experimental, transparencia computacional, sensibilidad a los supuestos del modelo y trazabilidad de los resultados. Por ello, los capítulos no se limitan a la presentación de modelos, sino que detallan los procedimientos metodológicos esenciales: cómo se definen las instancias de prueba, cómo se mide el desempeño de las soluciones, cómo se reportan los resultados de manera imparcial y cómo se evitan conclusiones sesgadas por configuraciones *ad hoc* o por la omisión de métodos de referencia sólidos. En este marco, la reproducibilidad se asume como una condición *sine qua non* para la credibilidad científica y la transferencia efectiva del conocimiento a la práctica.

De manera paralela, el libro aborda una visión tecnológica que está ganando terreno tanto en la investigación como en la industria: la integración de técnicas de búsqueda avanzada (como ALNS y otras metaheurísticas) con el soporte del aprendizaje automático (Machine Learning) para refinar las decisiones internas del solucionador. Esta línea de trabajo se presenta desde una perspectiva pragmática, no como una tendencia efímera. Se propone utilizar el aprendizaje automático, cuyo valor es innegable —por ejemplo, en la selección de operadores, la priorización de tareas o la estimación de riesgo—, mientras se mantiene un control operativo estricto mediante reglas de negocio, verificación de factibilidad y monitoreo constante de los niveles de servicio.

Finalmente, esta obra busca tender un puente entre dos audiencias que con frecuencia operan de forma aislada. Para la comunidad de investigadores, ofrece una estructura metodológica robusta y criterios de evaluación que fortalecen la validez del modelado y de la experimentación, incrementando la confiabilidad de las conclusiones sobre desempeño, robustez y capacidad de generalización. Para los profesionales de la implementación en empresas, proporciona una guía concreta sobre los datos mínimos requeridos, la instrumentación operacional crítica y las reglas de negocio, complementada con una lectura gerencial de las complejas compensaciones (trade-offs) entre costo, servicio y confiabilidad para facilitar la toma de decisiones estratégicas. Si este modelo contribuye a que un modelo sea más replicable, que un experimento sea más justo, o que una organización adopte el enrutamiento con mayor confianza y trazabilidad, su propósito se habrá cumplido.

Resumen

Este libro presenta una actualización metodológica y aplicada del Problema de Ruteo de Vehículos (VRP) con ventanas de tiempo, incorporando extensiones orientadas a la operación real, como la incertidumbre, el ajuste dinámico de rutas y enfoques de solución combinados. El texto parte de un modelo matemático base modernizado —formulado sobre una red de grafo, con restricciones de capacidad y ventanas de tiempo flexibles con holguras— e integra supuestos de incertidumbre operacional (demanda, tiempos de servicio y tiempos de viaje) tratados como variables aleatorias, junto con alternativas de modelado y criterios explícitos de robustez. A partir de esta base, se desarrolla un marco de evaluación reproducible que prioriza la comparabilidad experimental: la definición de instancias y bancos de prueba realistas, un protocolo estandarizado de comparación, métricas de calidad, de tiempo y de servicio, y recomendaciones actuales para el ajuste de parámetros y el reporte transparente. En el componente algorítmico, el libro organiza los métodos por familias —heurísticas, metaheurísticas, ALNS y enfoques híbridos ligeros— y enfatiza buenas prácticas para construir solucionadores comparables, en lugar de implementaciones aisladas y difíciles de contrastar. Asimismo, se discute la reciente línea de integrar el aprendizaje automático como apoyo a las decisiones internas del solucionador (selección de operadores, inserciones y reglas de priorización), manteniendo el control sobre la factibilidad y la trazabilidad. Finalmente, la evidencia experimental se traduce en implicaciones prácticas: una lectura gerencial de las compensaciones entre costo, servicio y confiabilidad, una guía de implementación empresarial (datos mínimos, telemetría, reglas de negocio y niveles de servicio) y una agenda de investigación emergente que articula escalabilidad, sostenibilidad, automatización y equidad de servicio. En conjunto, la obra busca tender un puente entre el rigor metodológico de la investigación y la adopción efectiva de soluciones de ruteo en contextos organizacionales.

Introducción

La logística de distribución se ha transformado en un campo donde conviven, en una tensión permanente, la optimización matemática y la realidad operativa. La expansión del comercio electrónico, la consolidación de compromisos de entrega en franjas horarias, la congestión urbana y la creciente disponibilidad de datos en tiempo real han alterado la naturaleza del Problema de Enrutamiento de Vehículos (PEV) (o VRP, por sus siglas en inglés).

En la práctica, la planificación de rutas ya no consiste solo en minimizar la distancia o el costo. Ahora implica sostener niveles de servicio medibles, responder a sucesos imprevistos, gestionar la incertidumbre en tiempos de viaje y de servicio, y garantizar la trazabilidad de las decisiones en sistemas que operan bajo una fuerte presión reputacional y contractual. Esta transformación exige un tratamiento metodológico que preserve el rigor de la formulación clásica, pero incorpore criterios contemporáneos de evaluación, solidez (robustez) y transferibilidad a contextos empresariales (Vidal *et al.,* 2020; Jastrząb *et al.,* 2024).

A pesar de la extensa tradición del VRP en la investigación de operaciones, persiste una brecha significativa entre los resultados académicos y su adopción efectiva. La transferencia a entornos reales se ve a menudo comprometida por las diferencias en los supuestos operativos, las exigencias de trazabilidad y la falta de alineación de los estándares de evaluación entre las comunidades (Accorsi *et al.,* 2022; Merchán *et al.,* 2024).

Una parte de esta divergencia se debe a la distancia entre los supuestos simplificadores y las condiciones reales: los plazos de entrega, que en la operación tienen flexibilidad y tolerancias; las demandas y los tiempos de servicio, que varían; los cambios de último momento en el conjunto de clientes; y las dinámicas de tráfico dependientes del tiempo.

Otra parte se debe a prácticas experimentales insuficientemente estandarizadas: comparaciones con líneas base (métodos de referencia) débiles, ausencia de control del presupuesto computacional, falta de replicaciones con distintas semillas (condiciones iniciales) o reportes incompletos del entorno de cómputo (Gunawan *et al.,* 2021; Jastrząb *et al*., 2024). En los enfoques recientes basados en aprendizaje automático, estas dificultades pueden amplificarse si no se separan adecuadamente las instancias de entrenamiento y de prueba, si se omite el costo de entrenamiento o de inferencia (predicción), o si se reportan métricas sin un análisis de dispersión y de generalización (Accorsi *et al.,* 2022).

Por ello, el avance actual del campo no solo depende de proponer nuevos algoritmos, sino también de fortalecer la credibilidad experimental y la interpretación de los resultados en términos de decisiones operativas. Este modelo busca responder a esas tensiones desde una lógica de actualización responsable.

Su punto de partida es un modelo base reformulado y modernizado para el VRP, con plazos de entrega y holguras, en el que la representación en grafo, las restricciones y la función objetivo se establecen con claridad y consistencia, evitando ambigüedades que obstaculizan la replicación. A partir de ese núcleo, el texto integra la incertidumbre operacional —en demanda, tiempos de servicio y tiempos de viaje— como un componente explícito, proponiendo supuestos y alternativas de modelado que permiten transitar de esquemas deterministas a enfoques estocásticos, sólidos o basados en escenarios. La motivación es reconocer que la calidad de una solución no puede juzgarse solo por su costo promedio: es necesario medir la estabilidad, los riesgos de colas de tardanza, las tasas de factibilidad bajo diversos escenarios y el cumplimiento de los Acuerdos de Nivel de Servicio (ANS) para clientes prioritarios.

En coherencia con lo anterior, la contribución principal del volumen es metodológica y aplicada: construir un marco de evaluación reproducible que permita comparar enfoques de manera equitativa y transferible. Se plantea un protocolo experimental con instancias y bancos de prueba de referencia que privilegia escenarios realistas por encima de ajustes *ad-hoc*, define métricas que reflejan simultáneamente calidad, tiempo, servicio y solidez, y enfatiza las buenas prácticas de transparencia computacional.

Este enfoque asume que la evidencia empírica debe sostenerse en condiciones presupuestarias equivalentes, con replicaciones suficientes y un reporte completo del *hardware* (equipo físico), de los parámetros y de las configuraciones. La comparabilidad también se entiende como un requisito para la lectura gerencial: solo con métricas claras y trazables es posible traducir resultados en decisiones como el nivel de tolerancia a la tardanza, la política de priorización, la densidad de ajuste dinámico de rutas o la inversión en capacidades tecnológicas.

El componente algorítmico se presenta clasificado por familias y arquitecturas, dando prioridad a la identificación de componentes reutilizables sobre la mera imitación de implementaciones específicas. Desde esta perspectiva, las heurísticas constructivas y las prioridades por plazos se interpretan como políticas operativas de despacho. Las metaheurísticas, que incluyen tanto los métodos de trayectoria como los poblacionales, se discuten en función de sus mecanismos de exploración y de sus respectivos vecindarios (Houssein *et al.,* 2024). Además, el marco de la Búsqueda Adaptativa en Vecindarios Grandes (Adaptive Large Neighborhood Search, ALNS) se presenta como una estructura moderna, especialmente adecuada para variantes complejas del problema, debido a su notable capacidad para combinar procesos de destrucción y reparación con la adaptación dinámica de operadores.

En paralelo, las metaheurísticas ligeras se abordan como estrategias pragmáticas para combinar la Programación Lineal Entera Mixta (PLEM) con heurísticas de forma localizada, cuando la operación requiere mejoras puntuales con restricciones fuertes o con un componente de certeza parcial. Sobre esta base, se introduce la tendencia contemporánea de la optimización aumentada por aprendizaje, no como un reemplazo total del solucionador, sino como una forma de mejorar las decisiones internas —selección de operadores, estimación de costos incrementales, priorización bajo riesgo—, manteniendo el cinturón de seguridad heurístico mediante la verificación de factibilidad, la reparación y la auditoría de decisiones.

El hilo conductor del modelo es, por tanto, la integración entre la formulación, la solución y la evaluación bajo criterios contemporáneos de solidez, reproducibilidad y transferibilidad a la operación. La obra incorpora una guía de implementación basada en datos mínimos, instrumentación operacional, reglas de negocio y ANS; discute las compensaciones costo–servicio–confiabilidad como decisiones gerenciales; y cierra con líneas de investigación emergentes (multiproducto, sostenibilidad, transición a PEV de Vehículos Eléctricos - EVRP, automatización del ajuste de parámetros y del reporte, equidad de servicio y arquitecturas con ajuste dinámico de rutas y gemelos digitales).

En síntesis, esta obra argumenta que la contribución más valiosa en el ruteo de vehículos contemporáneo no es únicamente reducir un porcentaje de costo en una batería de pruebas estandarizada, sino en construir evidencia y mecanismos que permitan decidir con confianza en la operación: qué se optimiza, con qué tolerancias, qué se sacrifica para sostener un ANS, qué tan sólido es el desempeño cuando el mundo real se aparta del supuesto determinista y, sobre todo, cómo auditar y reproducir tanto el experimento como la decisión. Bajo esa premisa, el libro se concibe como un puente entre el rigor académico y la transferencia tecnológica, aspirando a un estándar de reporte y diseño experimental y, simultáneamente, útil para la implementación empresarial.

Capítulo 1. El problema hoy: ruteo bajo incertidumbre y servicio diferenciado

1.1. Logística contemporánea y presión por nivel de servicio (última milla, tiempos, penalizaciones)

En la logística contemporánea, el ruteo de vehículos dejó de ser un ejercicio "estático" para convertirse en una decisión operativa sometida a variaciones permanentes: fluctuaciones del tráfico, patrones de demanda inestables, restricciones urbanas, cambios en la disponibilidad de clientes y exigencias de servicio cada vez más estrictas. La "última milla" intensifica estas tensiones porque concentra la incertidumbre y los costos en un espacio donde pequeños desajustes se traducen rápidamente en incumplimientos perceptibles para el usuario final (Boysen *et al*., 2021; Escudero-Santana *et al.,* 2022). En el contexto del comercio electrónico, por ejemplo, un elemento tan cotidiano como la presencia o la incertidumbre del cliente en el domicilio condiciona la programación y el ruteo, lo que empuja a modelos que integren explícitamente la probabilidad de fallas de entrega y sus consecuencias operativas (Özarık *et al*., 2021).

Este escenario ha consolidado una lógica de gestión basada en el nivel de servicio: no basta con minimizar los kilómetros o el tiempo total; ahora es necesario cumplir con ventanas de tiempo, reducir las tardanzas, evitar reintentos y mantener una experiencia consistente. Por ello, el desempeño se mide cada vez más mediante métricas que combinan eficiencia y confiabilidad (Soeffker *et al.,* 2022). En paralelo, emergen modalidades de entrega y recursos tecnológicos (como drones para ciertos segmentos de última milla) que amplían el espacio de decisiones, pero también agregan capas de restricciones, coordinación e incertidumbre (Jazairy *et al*., 2025). Es decir, la presión competitiva no solo exige rutas "más baratas", sino también rutas más seguras ante la variación, capaces de priorizar a los clientes, absorber perturbaciones y mantener los estándares de servicio.

Desde la investigación de operaciones, esta transición se manifiesta en el paso a problemas de ruteo más "realistas": dinámicos, estocásticos, dependientes del tiempo, con flota heterogénea y reglas de servicio diferenciado. Las revisiones recientes sobre ruteo dependiente del tiempo muestran cómo el campo ha avanzado en modelación y algoritmos en respuesta a redes viales con velocidades variables y congestión (Adamo et al., 2024), mientras que los estudios de ruteo dinámico resaltan que las decisiones deben reconfigurarse a medida que llega nueva información, cambiando el énfasis desde soluciones únicas hacia políticas de ajuste dinámico de rutas (Ojeda Rios *et al.,* 2021).

En términos operacionales, esta transición implica que el desempeño de un plan de rutas debe interpretarse como una combinación de eficiencia y cumplimiento del servicio bajo variación. En particular, los acuerdos de nivel de servicio (SLA) funcionan como un marco contractual y reputacional que traduce la "calidad del ruteo" en indicadores verificables: cumplimiento de ventanas, tardanza por percentiles, frecuencia de reintentos, estabilidad ante cambios y capacidad de priorizar segmentos críticos. En este tipo de operación, una solución que es barata en promedio puede ser inaceptable si falla sistemáticamente en escenarios de congestión o frente a incertidumbre en la atención o el servicio, lo que obliga a evaluar no solo el costo esperado, sino también su dispersión y su riesgo operativo (Soeffker *et al*., 2022; Adamo et al., 2024).

Además, el entorno tecnológico amplía el espacio de decisión: la telemetría, la disponibilidad de eventos en línea y la incorporación de nuevos recursos (por ejemplo, modalidades de entrega especializadas en ciertos contextos) refuerzan la necesidad de modelos que no se limiten a producir una ruta única, sino que soporten ajustes razonables del plan cuando cambia la información. Por ello, la formulación del problema y la evaluación de métodos deben alinearse con una lógica de operación real: decisiones auditables, métricas coherentes con el servicio y evidencia comparativa que sea defendible tanto en investigación como en implementación (Özarık *et al.,* 2021; *Jazairy et al*., 2025).

1.2. Formulación conceptual del problema base del modelo

El marco base que guía este modelo corresponde a un VRP enriquecido (VRP con múltiples restricciones) orientado a la operación real bajo incertidumbre. En adelante, lo denominamos: Modelo VRP-UF: *VRP de entregas bajo incertidumbre, con flota heterogénea, ventanas de tiempo flexibles y clientes prioritarios, y* tiempos de viaje dependientes del tiempo y de la reutilización de vehículos (multi-trip).

Para efectos de lectura y comunicación, en este modelo se utiliza la etiqueta VRP-UF como denominación operativa y sintética del problema base. No obstante, este modelo es consistente con el enfoque propuesto en la investigación bajo la sigla SFMVRPMSTWTDCP, que enfatiza explícitamente cinco rasgos: incertidumbre (demanda/servicio/viaje), flexibilidad operativa, múltiples ventanas blandas, tiempos de viaje dependientes del tiempo y priorización de entregas o de cobertura según criterios de servicio. En consecuencia, cuando el texto hable del VRP-UF, se hará referencia al mismo núcleo conceptual del enfoque, privilegiando la claridad sin perder la precisión técnica.

Conceptualmente, el VRP-UF se caracteriza por:

- *Demandas y tiempos estocásticos:* tanto la demanda/volumen de servicio como los tiempos de servicio y/o desplazamiento presentan variabilidad no controlable. Esto implica que la factibilidad y el costo de una ruta no son deterministas, y la solución debe ser evaluada por su desempeño esperado y su estabilidad ante perturbaciones (Soeffker *et al.,* 2022; Ojeda Rios *et al*., 2021).
- *Tiempos de viaje dependientes del tiempo (time-dependent):* la duración de los arcos no es constante; depende de la hora (congestión pico/valle), de restricciones urbanas y de condiciones operativas. En consecuencia, el orden de visita y el momento de salida afectan de manera no lineal el cumplimiento de las entregas (Adamo *et al.,* 2024).
- *Flota heterogénea*: los vehículos difieren en capacidad, costos y desempeño. Esto modifica la asignación óptima porque cada ruta "cuesta" distinto según el tipo de vehículo y su adecuación a la demanda y al territorio. Además, la heterogeneidad es un rasgo natural de operaciones reales (combinación de motos, vans, camiones livianos, etc.) y se vuelve aún más relevante cuando se consideran alternativas tecnológicas o energéticas (Küçükoğlu *et al.,* 2021).
- *Ventanas de tiempo flexibles (soft time windows):* en lugar de prohibir totalmente la tardanza o la anticipación, se permiten desviaciones con una penalización. Este enfoque refleja cómo operan muchas cadenas logísticas: el incumplimiento no siempre invalida la solución, pero sí genera costos (compensaciones, deterioro del SLA, reintentos y reputación). Los modelos y análisis de ruteo con ventanas suaves evidencian la necesidad de incorporar penalizaciones para capturar de manera realista el "costo del servicio" (Xu *et al.,* 2020).

- **Clie***ntes prioritarios (prioridades de servicio):* no todos los clientes tienen el mismo peso. Algunos cuentan con acuerdos de servicio más estrictos, de mayor valor, de mayor criticidad sanitaria o industrial, o de mayor impacto reputacional. Esto obliga a introducir reglas de priorización y esquemas de penalización diferenciados, especialmente en la última milla con alta incertidumbre de presencia/atención (Özarık *et al*., 2021).
- Reutilización de vehículos (multi-trip / múltiples rutas por jornada): un mismo vehículo puede completar una ruta, retornar (o reabastecerse) y ejecutar otra en la misma jornada. Este rasgo conecta el ruteo con restricciones de turno, horas extra y capacidad operativa, y es clave cuando se busca flexibilidad para atender variabilidad diaria (Ojeda Rios et al., 2021; Soeffker *et al.,* 2022).

En conjunto, el VRP-UF se interpreta como una decisión operativa de asignación y secuenciación de visitas que debe equilibrar el costo total, el cumplimiento de servicio y la robustez. Metodológicamente, esta complejidad hace que los enfoques exactos se limiten a tamaños pequeños y que, en escalas realistas, se recurra a heurísticas y metaheurísticas modernas, entre las que destacan marcos como ALNS y sus operadores (Mara *et al*., 2022; Voigt, 2025).

En términos de decisión, el problema combina (i) asignación de clientes y vehículos, (ii) secuenciación de visitas, (iii) programación temporal (salidas/llegadas) y (iv) manejo de la incertidumbre mediante escenarios o distribuciones, lo que vuelve natural que el desempeño deba interpretarse como una distribución (y no un valor único). Esta lectura es especialmente relevante cuando se incorporan prioridades, ya que el costo total puede disminuir a expensas de degradar el servicio en segmentos críticos, o viceversa, lo que hace necesario explicitar el criterio de equilibrio en la función objetivo (Xu *et al*., 2020; Özarık *et al*., 2021).

1.3. Componentes de costo total (fijo, variable, espera, penalizaciones, horas extra)

Según el planteamiento del modelo, el desempeño de una solución no puede evaluarse únicamente por la distancia. Se requiere un costo total que represente adecuadamente lo que una operación paga (en términos de dinero, tiempo y servicio). Conceptualmente, el costo total se compone de:

1. **Costo fijo de uso de vehículo**: costo asociado a habilitar un vehículo (o un tipo de vehículo) para operar, incluyendo la disponibilidad, el mantenimiento básico o la contratación. Este componente es central en flotas heterogéneas, donde la elección del tipo de vehículo forma parte de la estrategia (Nucamendi-Guillén *et al.,* 2021).
2. **Costo variable de transporte**: depende del recorrido, del tiempo de conducción y, en escenarios dependientes del tiempo, de la franja horaria en la que se transita. La literatura sobre ruteo dependiente del tiempo enfatiza que el costo operativo se ve afectado por la congestión y la dependencia temporal de los arcos (Adamo *et al*., 2024).
3. **Costo de espera (ociosidad)**: ocurre cuando el vehículo llega antes de la ventana de tiempo y debe esperar para ser atendido. Aunque a veces reduce las tardanzas, incrementa los tiempos improductivos y puede afectar la capacidad de completar múltiples rutas en un mismo día (Xu *et al*., 2023; Gutiérrez *et al*., 2024).
4. **Penalizaciones por violación de ventanas (soft time windows)**: cuando se permite la tardanza o la anticipación, la penalización modela el deterioro del nivel de servicio. En este punto, las contribuciones sobre ventanas suaves indican que la penalización es una forma de internalizar el impacto del incumplimiento en la función objetivo (Xu *et al.,* 2020).
5. **Horas extra / overtime**: si el plan excede la jornada o los límites de operación, el costo aumenta. Este elemento cobra relevancia cuando se admite la reutilización de la flota y cuando el sistema reacciona ante la incertidumbre, pues la "robustez" a menudo compite con la restricción temporal del turno (Soeffker *et al*., 2022).
6. **Costos por fallas de entrega (cuando aplica)**: en la última milla, la incertidumbre sobre la presencia del cliente conduce a reintentos o reprogramaciones. Aunque no siempre se

modela explícitamente en todos los VRP, la evidencia muestra que esta dimensión es operacionalmente determinante en el e-commerce y debe representarse de alguna forma (Özarık *et al*., 2021).

Esta estructura de costos prepara el terreno para una idea clave del modelo de investigación: en el ruteo de entregas bajo incertidumbre, lo 'óptimo' no es solo eficiencia, sino un compromiso explícito entre costo total y nivel de servicio, determinado por el diseño de penalizaciones y las reglas de priorización.

Desde el punto de vista metodológico, esta desagregación permite que el análisis no se reduzca a un número final, sino que identifique qué componente explica la mejora o el deterioro (por ejemplo, la reducción de la distancia a costa de una mayor espera o de una mayor tardanza penalizada). Además, cuando hay incertidumbre, estos costos deben reportarse no solo como promedio, sino también con medidas de dispersión (p. ej., desviación estándar o percentiles) para distinguir entre soluciones eficientes pero frágiles y soluciones más estables ante perturbaciones. En particular, el costo de espera —aunque puede disminuir la tardanza— puede incrementar la improductividad y afectar la capacidad operativa para completar múltiples rutas; por ello, conviene observarlo explícitamente junto con las penalizaciones de servicio (Xu et al., 2023; Gutiérrez *et al*., 2024; Xu *et al.,* 2020).

1.4. Preguntas de investigación y contribución (robustez, confiabilidad, replicabilidad)

Para alinear el libro con estándares actuales (y no con una lógica únicamente descriptiva), planteamos preguntas que organizan la discusión y justifican decisiones metodológicas:

P1. ¿Cómo cambia la calidad del ruteo cuando el tiempo de viaje depende del tiempo y, además, es estocástico?

Esta pregunta conecta el ruteo time-dependent con la incertidumbre: rutas que son buenas "en promedio" pueden fallar con frecuencia si no se evalúa la variabilidad (Adamo *et al*., 2024; Soeffker *et al*., 2022).

P2. ¿Qué esquema de ventanas flexibles y penalizaciones permite equilibrar costo y servicio sin convertir el modelo en una regla arbitraria?

Aquí se busca fundamentar el uso de soft time windows como una representación realista del servicio y analizar cómo el peso de las penalizaciones altera las decisiones de secuenciación y asignación (Xu et al., 2020).

P3. ¿Cómo incorporar un servicio diferenciado (clientes prioritarios) sin destruir la eficiencia global?

La priorización introduce asimetrías en la función objetivo; el reto es diseñar un esquema que garantice el servicio crítico (o de alto valor) sin provocar costos desproporcionados en el resto del sistema (Özarık *et al*., 2021).

P4. ¿Qué familia de métodos ofrece el mejor equilibrio entre la calidad, el tiempo de cómputo y la estabilidad del resultado en el Rich VRP?

Las revisiones sobre heurísticas modernas muestran una convergencia hacia marcos adaptativos (ALNS) y diseños de operadores más sistemáticos para VRP ricos (Mara *et al.,* 2022; Voigt, 2025). El libro adopta esta discusión y la contrasta con otras familias metaheurísticas, sin depender de resultados únicos, sino de evidencia comparativa.

P5. ¿Cómo asegurar la confiabilidad y la replicabilidad en la evaluación de algoritmos, especialmente cuando se incorporan componentes de aprendizaje automático?

La evaluación de métodos con ML requiere protocolos de prueba más estrictos (definición de instancias, control de semillas, métricas, reporte de cómputo) para evitar conclusiones frágiles. Las guías para pruebas computacionales en este contexto constituyen una referencia directa para modernizar el enfoque (Accorsi *et al*., 2022; Bai *et al.,* 2023). Además, la visión metodológica sobre el ML en optimización combinatoria ayuda a encuadrar lo que el aprendizaje realmente puede aportar a estos problemas (Bengio *et al.,* 2021).

Estas preguntas no se plantean como un inventario teórico, sino como un criterio práctico de diseño: cada una exige decisiones explícitas en (i) la formulación (ventanas flexibles, estructura de penalizaciones, prioridad), (ii) la evaluación (métricas que capturen costo–servicio–estabilidad) y (iii) el protocolo experimental (control de presupuesto computacional, replicaciones y reporte transparente). En ese sentido, la contribución metodológica se fundamenta en la idea de "comparación justa": si dos métodos se comparan, deben hacerlo bajo condiciones equivalentes y con indicadores consistentes con lo que la operación realmente paga y arriesga (Accorsi *et al.,* 2022; Bai *et al.,* 2023).

Contribución del modelo: proponer un marco integrado de modelación y solución para el ruteo bajo incertidumbre y el servicio diferenciado, con evaluación reproducible y comparaciones metodológicamente defendibles.

1.5. Estructura del modelo de investigación y orientación metodológica.

El modelo está organizado para servir a dos públicos, manteniendo la solidez metodológica y la aplicabilidad operativa:

Para lectores académicos (investigación/posgrado):

- En el Capítulo 1 se establecen los fundamentos y generalidades: motivación del problema, alcance, supuestos operativos y el marco conceptual que conecta la formulación, la solución y la evaluación.
- En el Capítulo 2 se presenta el estado del arte (con énfasis en la literatura reciente y en líneas de investigación de ALNS, VRP dinámico-estocástico y ML/analítica en ruteo).
- En el Capítulo 3 se formalizan el modelo y el diseño experimental, con foco en la replicabilidad.
- En el Capítulo 4 se discuten métodos de solución, con comparaciones metodológicas basadas en revisiones recientes y en buenas prácticas de evaluación (Accorsi **et al**., 2022; Mara *et al*., 2022; Voigt, 2025; Bai *et al*., 2023).

Para lectores practicantes (operación/ingeniería logística):

- El Capítulo 1 (este) clarifica el problema operacional y las fuentes de variación.
- El Capítulo 5 ofrece una interpretación gerencial: cómo elegir penalizaciones, cómo justificar prioridades, cómo decidir la mezcla de flota y cómo preparar datos y operación para el ajuste dinámico de rutas y de los escenarios de última milla (Özarık *et al.,* 2021; Jazairy *et al.,* 2025).

Con esta ruta, el lector puede leer el libro "en línea" (1→5) o usarlo de manera modular: por ejemplo, quien implemente soluciones puede ir de 1 a 5 y consultar 3–4 según la necesidad; quien investigue puede profundizar en 2–4 y usar 5 como agenda.

Con esta organización, el documento busca facilitar dos lecturas complementarias: una lectura explicativa, orientada a comprender supuestos, formulación y evidencia; y una lectura instrumental, orientada a implementar (datos mínimos, reglas de negocio y medición del nivel de servicio) y a justificar decisiones ante actores internos. En ambos casos, la intención es que el lector pueda rastrear sin ambigüedad qué se asumió, qué se midió y cómo se llegó a una conclusión, de modo que la transferencia a la operación no dependa de interpretaciones implícitas, sino de criterios y procedimientos verificables (Özarık *et al.,* 2021; Jazairy *et al.,* 2025).

Capítulo 2. Estado del arte: rutas, incertidumbre, y solución híbrida

2.1. De VRP "clásico" a VRP "real": incertidumbre, dinamismo y servicio

La investigación reciente coincide en que el VRP moderno ya no se entiende como una sola decisión óptima "offline", sino como un proceso secuencial de decisiones sometidas a nueva información (demandas, presencia del cliente, congestión, eventos), donde el objetivo operativo real se desplaza de "minimizar distancia" a garantizar servicio con costo controlado (Ojeda Rios *et al.,* 2021; Soeffker *et al*., 2022). Este cambio ha impulsado una taxonomía más rica: VRP dinámico (DVRP) cuando aparecen eventos en línea; VRP estocástico cuando los parámetros son aleatorios; y VRP estocástico-dinámico cuando ambas condiciones coexisten, lo que exige políticas (o ajustes dinámicos del plan de rutas) y no solo rutas (Ulmer, 2020; Soeffker et al., 2022).

En paralelo, la literatura señala que el nivel de servicio actúa como un "segundo objetivo" que, en la práctica, se incorpora mediante penalizaciones (SLA, tardanzas) y reglas de priorización, en particular en última milla (Özarık *et al.,* 2021). Por eso, los enfoques contemporáneos tienden a trabajar con modelos "rich", en los que la calidad se mide por el costo total, la confiabilidad y la capacidad de respuesta. Esta visión se conecta con la línea de analítica prescriptiva: integrar modelos de información (predicción, anticipación) con modelos de decisión (optimización) para tomar mejores decisiones bajo incertidumbre (Soeffker *et al*., 2022).

En términos operativos, esta transición implica que el "mismo" plan de rutas debe poder sostenerse ante desviaciones inevitables: variaciones en la duración real del servicio, congestiones no anticipadas, cancelaciones y reprogramaciones. Por eso, en VRP bajo incertidumbre, la comparación entre métodos no puede reducirse al mejor costo observado: debe incorporar confiabilidad (qué tanto se mantiene el servicio), robustez (qué tanto se degrada el desempeño cuando cambian las condiciones) y capacidad de reacción (qué tan bien se ajustan las decisiones ante nueva información). Esta lectura conecta directamente el estado del arte con tu necesidad metodológica: evaluar soluciones en función del desempeño en escenarios y no como una única cifra "óptima".

2.2. Incertidumbre operacional y time-dependent routing: el tráfico ya es parte del modelo

Un avance fuerte, cada vez más adoptado como condición de realismo en la literatura reciente, es tratar los tiempos de viaje como dependientes del tiempo: la red vial cambia con la hora, y la ruta óptima puede variar con el momento de salida, el orden de visita y la interacción con ventanas (Adamo *et al.,* 2024). Esta línea subraya que el ruteo urbano real incorpora comportamiento no lineal: un cambio pequeño en la secuencia puede provocar tardanzas si la ruta entra en hora pico.

Además, al combinar lo time-dependent con la incertidumbre (tráfico, incidentes), la evaluación deja de ser "un valor" y pasa a ser una distribución de desempeño (Ulmer, 2020; Soeffker *et al*., 2022). Por eso, la literatura impulsa enfoques que reporten el desempeño esperado, la variabilidad, las tasas de incumplimiento y la robustez, no solo el mejor costo.

Desde el punto de vista de la modelación, "dependiente del tiempo" no significa únicamente reemplazar una matriz fija por otra, sino reconocer que la decisión de secuenciar visitas y la de cuándo transitar están acopladas. Esto vuelve especialmente relevante (i) el comportamiento de las ventanas de tiempo cuando se cruza con la congestión pico/valle y (ii) el hecho de que las penalizaciones por tardanza pueden dispararse por efectos de umbral (entrar a una franja

crítica). En consecuencia, los modelos y algoritmos que pretenden ser transferibles a operación real necesitan explicitar cómo representan esa dependencia (perfil horario, escenarios, funciones por tramo) y cómo se mide su impacto sobre servicio y confiabilidad.

2.3. Soft time windows: penalizar para modelar servicio (no solo restringir)

Las ventanas flexibles (soft) son cruciales porque capturan lo que ocurre en operaciones reales: llegar tarde no hace la ruta "inviable", pero sí genera costos, reclamos o reintentos. En esa lógica, la penalización se convierte en un "precio" del servicio. Trabajos recientes en VRP con ventanas flexibles discuten cómo el modelado y la calibración de estas penalizaciones afectan las decisiones de asignación y secuenciación, y permiten explorar trade-offs de forma controlada (Xu *et al.,* 2020).

Lo relevante para el estudio es que las soft windows habilitan la integración coherente de: (a) la incertidumbre, (b) la prioridad de los clientes y (c) el ajuste dinámico de rutas, porque permiten absorber desviaciones sin romper el modelo.

En términos metodológicos, las ventanas flexibles también abren la puerta a comparaciones más justas: dos métodos pueden presentar costos similares, pero con perfiles de tardanza muy distintos. Por eso, la literatura reciente tiende a tratar la tardanza no como "fallo binario", sino como una variable de desempeño que debe resumirse mediante métricas robustas (promedios, percentiles y dispersión) y, cuando existen prioridades, mediante métricas segmentadas. Esta idea se conecta con el Capítulo 5: las penalizaciones no son un artificio matemático, sino el mecanismo que traduce una promesa de servicio en una decisión optimizable y auditable.

2.4 Servicio diferenciado y última milla: presencia incierta, reintentos y priorización

La última milla añade un componente que cada vez aparece más en la literatura: la incertidumbre sobre la presencia del cliente. En el e-commerce, no basta con "llegar dentro de la ventana"; existe la probabilidad de falla (no atención, no recepción), lo que obliga a tomar decisiones de programación y ruteo que consideren el riesgo y no solo el tiempo (Özarık *et al*., 2021).

Desde la perspectiva del servicio diferenciado, esto abre un marco natural: los clientes prioritarios pueden enfrentar penalizaciones más altas por tardanza o falla, así como reglas de recuperación más agresivas (reintento, reasignación). Este tema no se resuelve solo con "ponderaciones"; requiere consistencia entre (i) la función objetivo (penalizaciones y costos), (ii) la política de ajuste dinámico de rutas (cuándo replanificar) y (iii) los datos y estimaciones del entorno (Soeffker *et al*., 2022; Ojeda Rios *et al*., 2021).

Esta sección es especialmente importante porque conecta directamente el "servicio diferenciado" con decisiones típicas de empresa: qué clientes son críticos, cuánto vale evitar un reintento, qué nivel de tardanza es tolerable antes de activar un ajuste del plan, y qué métricas sirven para justificar esas decisiones ante SLAs. Dicho de otro modo: la priorización no solo define "a quién atender primero", sino también cómo se redistribuye el riesgo operacional cuando el día real se aparta del plan.

2.5. Métodos dominantes en la literatura reciente: ALNS y los operadores como unidad de diseño

En VRP con múltiples restricciones (VRP “enriquecidos”), la evidencia reciente se ha consolidado en torno a metaheurísticas adaptativas, en particular, Adaptive Large Neighborhood Search (ALNS), por su flexibilidad para combinar esquemas de destrucción/reparación, manejar múltiples restricciones y adaptarse a distintas variantes del VRP dentro de un mismo marco (Mara *et al.,* 2022). El interés contemporáneo se ha desplazado desde “usar ALNS” hacia el diseño y la comparación de operadores (qué destruir, cómo reparar, cómo puntuar y cómo adaptar), incluyendo rankings y análisis sistemáticos de operadores en VRP (Voigt, 2025).

Para el modelo, esto es clave: si el problema incluye tiempos de viaje variables según la hora, ventanas de tiempo flexibles, prioridades de servicio y múltiples recorridos por jornada, ALNS permite una construcción modular: cada rasgo puede incorporarse mediante penalizaciones, verificaciones de factibilidad o como parte del mecanismo de reconstrucción de la solución. En consecuencia, la literatura tiende a privilegiar no “un algoritmo” aislado, sino un esquema de solución capaz de integrar componentes sin tener que reescribir el método desde cero.

Esta evolución tiene implicaciones directas para la evaluación: si los operadores son “la unidad de diseño”, entonces el análisis no debería reportar únicamente el resultado final, sino también qué componentes explican el desempeño. De ahí que resulte relevante reportar configuraciones, reglas de aceptación, criterios de penalización y sensibilidad a parámetros, además de incorporar análisis por partes (por ejemplo, cómo cambia el desempeño al activar o desactivar familias de operadores). Esta perspectiva prepara el paso al diseño experimental del Capítulo 3 y evita que los resultados dependan de decisiones implícitas, difíciles de auditar.

2.6. Analítica y machine learning en VRP: de la promesa a la evaluación replicable

La literatura reciente no plantea el aprendizaje automático como sustituto directo de la optimización, sino como un componente de apoyo para mejorar la toma de decisiones en problemas combinatorios: aprender reglas de selección, aproximar costos difíciles, priorizar movimientos o guiar la búsqueda en espacios enormes donde una estrategia “manual” resulta costosa o poco generalizable. Este encuadre, ampliamente discutido desde la intersección entre la investigación de operaciones y el aprendizaje, enfatiza que el valor del ML depende de qué se aprende, con qué datos y cómo se integra con heurísticas y modelos de decisión (Bengio *et al.,* 2021). Sin embargo, un punto crítico (y muy alineado con tu exigencia de “contenido actualizado”) es que hoy se insiste en protocolos de prueba computacional: control de semillas, instancias, tiempo de cómputo, métricas coherentes y reportes comparables, especialmente cuando hay ML involucrado. En VRP, estas guías son centrales para evitar resultados “frágiles” o no reproducibles (Accorsi *et al.,* 2022). La investigación, por tanto, no puede limitarse a describir “qué funciona”, sino explicar cómo probarlo bien.

En el caso específico del ruteo, este punto se vuelve todavía más exigente: la evidencia acumulada muestra que los avances más robustos no provienen de “aprender rutas completas” en abstracto, sino de enfoques híbridos donde el aprendizaje apoya componentes de la solución (por ejemplo, selección de operadores, políticas de inserción, o parametrización), manteniendo un núcleo de optimización/heurística que asegura factibilidad y control del desempeño. Una síntesis actual y altamente útil para el libro es la encuesta en IEEE Transactions on Intelligent Transportation Systems, que organiza la literatura de ML aplicada a VRP en una taxonomía por paradigmas de aprendizaje, estructuras de solución e hibridaciones con métodos tradicionales, además de discutir ventajas y límites en términos de generalización y competitividad frente a heurísticas diseñadas a mano (Bogyrbayeva *et al.,* 2024).

En VRP, la discusión metodológica reciente insiste en protocolos transparentes de prueba computacional (instancias, semillas, tiempos, métricas y reportes) para evitar conclusiones frágiles, algo particularmente crítico cuando se incorporan componentes de aprendizaje (Accorsi *et al.,* 2022). En otras palabras, el aporte de ML en ruteo debe evaluarse con el mismo rigor que cualquier contribución algorítmica: no solo por el mejor resultado, sino también por la estabilidad, el costo computacional y la reproducibilidad.

En consecuencia, el aporte de ML en ruteo debe juzgarse con el mismo estándar que cualquier método: (i) costo computacional total (incluido el entrenamiento si aplica), (ii) estabilidad en escenarios y (iii) capacidad de generalizar a instancias distintas de las "favorables". Esto no reduce el valor del aprendizaje automático; lo encuadra donde suele aportar más: apoyar en decisiones internas (selección de operadores, estimación de riesgo, priorización) sin perder factibilidad ni trazabilidad.

2.7. Tendencias tecnológicas que reconfiguran el ruteo: EVRP y drones

Dos líneas tecnológicas afectan directamente el planteamiento del problema:
(i) Electrificación (EVRP y variantes). La literatura muestra una expansión del EVRP hacia variantes con recarga, restricciones energéticas y decisiones operativas que interactúan con ventanas de tiempo y con la planificación urbana. Un punto importante es que el EVRP no es solo "otro VRP": cambia el espacio de factibilidad y el rol del tiempo, por lo que demanda enfoques de modelación y de solución específicos (Küçükoğlu *et al*., 2021).

(ii) Drones y configuraciones híbridas. Los drones aparecen como una alternativa —no universal— para segmentos de última milla, con beneficios potenciales y restricciones regulatorias y operativas. Las revisiones sistemáticas recientes organizan esta evidencia desde la perspectiva de la gestión logística y evidencian su naturaleza interdisciplinaria (Jazairy *et al.,* 2025). Para este modelo, lo importante no es "optimizar drones", sino comprender cómo estas tecnologías empujan problemas más heterogéneos y multi-recurso.

Estas tendencias importan para tu modelo porque empujan operaciones con recursos diversos y restricciones distintas (energía, recarga, coordinación multi-recurso), lo cual refuerza la necesidad de (a) funciones de costo más completas (no solo distancia), (b) métricas de servicio segmentadas por criticidad y (c) protocolos experimentales que permitan comparar enfoques bajo supuestos explícitos.

2.8. Síntesis de brechas (gap) y posicionamiento del modelo propuesto de VRP

A partir del estado del arte, se observan cuatro brechas que justifican la propuesta del modelo:

1. Integración realista de rasgos: muchos trabajos abordan incertidumbre, time-dependent, soft windows o prioridad por separado; la operación real los integra (Adamo *et al*., 2024; Soeffker et al., 2022).
2. Servicio diferenciado bajo incertidumbre: el tratamiento de presencia incierta y priorización aún requiere marcos comparables y criterios gerenciales que conecten penalizaciones con decisiones (Özarık *et al*., 2021).
3. Arquitecturas de solución replicables: ALNS domina como marco, pero la replicabilidad exige protocolos transparentes de comparación, sobre todo al incorporar ML (Accorsi et al., 2022; Mara *et al.,* 2022; Voigt, 2025).
4. Actualización tecnológica: EVRP y modalidades emergentes (p. ej., drones) obligan a re-pensar flota heterogénea, energía y coordinación (Küçükoğlu *et al.,* 2021; Jazairy *et al*., 2025).
5. Brecha de operacionalización "data-to-decision": aun cuando se reconoce la necesidad de reoptimizar con información nueva, muchas formulaciones quedan desconectadas del ciclo

operativo real (captura de datos → actualización del estado → replanificación → ejecución → monitoreo). En logística urbana, el enfoque de Digital Twins aporta precisamente un marco para cerrar esa brecha: no solo modela el sistema, sino que estructura la integración entre datos, simulación y decisiones, lo que permite evaluar políticas y reacciones ante eventos con una lógica sistémica (Belfadel *et al.,* 2023).

En conjunto, estas brechas justifican un modelo de ruteo centrado en la operación real: flota heterogénea, ventanas flexibles, prioridades de servicio y variabilidad operacional, evaluado con un protocolo reproducible. Este posicionamiento define el objetivo del siguiente capítulo: formalizar el modelo y, sobre todo, fijar un diseño experimental que permita sostener comparaciones defendibles, tanto en términos académicos como en términos de decisión organizacional.

Con base en ello, el libro se posiciona como un marco integrado (modelo + solución + evaluación) para el ruteo bajo incertidumbre y servicio diferenciado, alineado con estándares contemporáneos de reproducibilidad y transparencia metodológica.

Capítulo 3. Modelo actualizado y diseño experimental reproducible

3.1. Modelo matemático base del modelo

Esta investigación adopta como base un modelo de ruteo sobre un grafo dirigido, formulado para capturar rutas, capacidad, tiempos de servicio y ventanas de tiempo, con mecanismos explícitos de flexibilidad y holgura, de modo que el modelo sea extensible a variantes dinámicas y estocásticas (Ulmer, 2020; Ojeda Rios *et al*., 2021; Adamo *et al*., 2024).

Estructura del grafo y conjuntos. Sea *G = (V, A)* un grafo dirigido con:

- $V = \{0\} \cup N \cup \{n+1\}$ donde 0 es el depósito de salida, n+1 puede representar el depósito de retorno y *N = {1,…,n}* el conjunto de clientes.
- $A = \{(i,j) \in V \times V : i \neq j\}$ el conjunto de arcos posibles.
- $K = \{1, \dots, m\}$ conjunto de vehículos

Para cada cliente $i \in N$; demanda q_i ; tiempo de servicio s_i y una ventana de tiempo "base" $[a_i, b_i]$. La distancia o costo de viaje en *(i,j)* se denota c_{ij} (o d_{ij}) y el tiempo de viaje t_{ij}. En versiones modernas t_{ij} puede ser dependiente de la hora de salida (time-dependent), es decir $t_{ij}(\tau)$, lo cual es clave para entornos urbanos (Adamo *et al*., 2024).

Variables de decisión (rutas)

- $x^k_{ij} \in \{0,1\}$: vale 1 si el vehículo *k* viaja directamente de *i* a *j*.
- $T^k_j \geq 0$: tiempo de inicio de servicio del vehículo *k* en el nodo *i*.
- $L^k_i \geq 0$: carga restante o carga acumulada del vehículo *k* al llegar a *i* (según convención).
- Variables de holgura/penalización para "ventanas blandas" (soft):
- $e^k_i \geq 0$: llegada temprana respecto a a_i (si se penaliza o si modela espera).
- $\ell \geq 0$: tardanza respecto a b_i (violación controlada).

Restricciones de cobertura y flujo. Cada cliente se visita exactamente una vez:

$$\sum_{k\in K}\sum_{J\in V\ j\neq i} x_{ij}^{k} = 1 \quad \forall i \in N$$

Conservación de flujo para cada vehículo:

$$\sum_{J\in V\ j\neq i} x_{ij}^{k} - \sum_{J\in V\ j\neq i} x_{ij}^{k} = 0 \quad \forall i \in N, \forall k \in K$$

Salida del depósito y retorno (si aplica):

$$\sum_{J\in V\ j\neq i} x_{0j}^{k} \leq 1, \sum_{i\in V\ i\neq n+1} x_{i(n+1)}^{k} \leq 1 \quad \forall k \in K$$

Capacidad. Con capacidad *Q* por vehículo, usando un esquema tipo MTZ o "carga acumulada":

$$L_j^k \geq L_I^k + q_j - M\left(1 - x_{ij}^k\right) \quad \forall (i,j) \in A, \forall k \in K$$
$$0 \leq L_i^k \leq Q \ \forall i \in V, \forall k \in K$$

Propagación temporal y ventanas de tiempo. Si el vehículo *k* recorre *(i,j)* debe iniciar servicio en *j* después de viajar y servir en *i:*

$$T_j^k \geq T_i^k + s_i + t_{ij}\left(T_i^k\right) - M\left(1 - x_{ij}^k\right) \quad \forall (i,j) \in A, \forall k \in K$$

Ventana blanda (soft), usando holguras:

$$T_i^k \geq a_i - e_i^k, \quad T_i^k \leq b_i - l_i^k \quad \forall i \in N, \forall k \in K$$

Este mecanismo permite reflejar realismo operacional: no todas las violaciones son imposibles, pero sí tienen costo, lo que es coherente con modelos modernos de atención al cliente, presencia incierta o promesas de entrega (Xu *et al.,* 2020; Özarık *et al.,* 2021).

Ventanas flexibles y holguras de ajuste. En muchos entornos la ventana $[a_i, b_i]$ no es fija: puede moverse dentro de un rango permitido por la política de servicio (flexibilidad negociada) o por tolerancias (Ulmer, 2020). Una forma práctica es incorporar un "desplazamiento" controlado:

$\delta_i^- \geq 0$: $extensión\ hacia\ antes\ del\ inicio\ efectivo$,
$\delta_i^+ \geq 0$: $extensión\ hacia\ después\ del\ final\ efectivo$,
Con límites $\delta_i^- \leq \overline{\delta}_i^-$, $\delta_i^+ \leq \overline{\delta}_i^+$. Así:
$T_i^k \geq (a_i - \delta_i^-) - e_i^k$, $T_i^k \leq (b_i - \delta_i^+) + l_i^k$

Aquí aparecen dos capas de flexibilidad:

1. Flexibilidad contractual (mover la ventana con $\delta_i^{\pm}$), y
2. Holguras de violación (soft) e_i^k, l_i^k

Esto hace al modelo más "operable" en contextos reales y prepara el puente hacia la incertidumbre operativa y hacia decisiones robustas (Xu et al., 2020; Soeffker *et al.,* 2022).

Para alinear el modelo con la operación real, se recomienda explicitar, desde esta formulación, dos extensiones estructurales que suelen quedar implícitas en descripciones informales: la heterogeneidad de la flota y la activación del vehículo. En flota heterogénea, la capacidad y el costo no son uniformes, por lo que es natural parametrizar por vehículo (o tipo de vehículo) mediante Q_k, costos fijos F_k y costos variables c^k_{ij} o t^k_{ij} (τ) cuando existan diferencias por desempeño o restricciones operativas. Complementariamente, una variable de uso $y_k \in \{0,1\}$ permite vincular la decisión de "abrir" un vehículo con el costo fijo, y evita que el modelo asuma disponibilidad ilimitada por defecto. Estas extensiones no alteran la lógica base (cobertura, flujo, capacidad y tiempo), pero hacen que el modelo sea consistente con los componentes de costo

total y con decisiones típicas de operación sobre mezcla de flota (Ulmer, 2020; Ojeda Rios et al., 2021).

3.2. Incertidumbre operacional: demanda/servicio/viaje como variables aleatorias (supuestos y alternativas)

En escenarios reales, la planificación se enfrenta a la variabilidad en la demanda, los tiempos de servicio y los tiempos de viaje. Un modelo reproducible no debe ocultar este hecho: debe declarar supuestos, justificar distribuciones o escenarios y mostrar alternativas metodológicas (Ulmer, 2020; Ojeda Rios *et al.,* 2021; Soeffker *et al*., 2022).

Fuentes típicas de incertidumbre:

- *Demanda aleatoria:* Q_i puede depender de comportamiento del cliente o abastecimiento parcial.
- *Servicio aleatorio:* S_i varía por preparación, verificación, pago o congestión en punto de entrega.
- *Viaje aleatorio / dependiente del tiempo:* $T_{ij}(\tau)$ cambia con la hora, congestión y eventos, lo que se conecta con VRP dependiente del tiempo (Adamo et al., 2024).
- *Presencia del cliente* (last-mile): probabilidad de "no show" o atención fallida (Özarık et al., 2021).

Supuestos mínimos recomendables (declarativos):

1. *Independencia o correlación:* definir si Q_i, S_i Son independientes entre clientes o presentan correlación espacial/temporal.
2. *Estacionariedad o no:* definir si el proceso de viaje es estable por franjas horarias o cambia según el día.
3. *Observabilidad:* qué se conoce al planificar (offline) y qué se revela en la operación (online/dinámico).
4. *Política de recourse:* qué acciones están permitidas cuando ocurre incertidumbre (ajuste dinámico de rutas, inserción dinámica, reasignación, rechazo).

Alternativas de modelado:

- *VRP dinámico-estocástico:* los parámetros se revelan en el tiempo; se decide mediante un ajuste dinámico de rutas y reglas de recourse (Ulmer, 2020; Soeffker et al., 2022).
- *Programación estocástica por escenarios:* se generan escenarios $\omega \in \Omega$ con probabilidad p_ω y se minimiza costo esperado; puede imponerse consistencia entre "primera etapa" (rutas base) y "segunda etapa" (ajustes).
- *Optimización robusta:* se definen conjuntos de incertidumbre (cajas, elipsoides) para q_j, t_{ij} y se minimiza el peor caso o un compromiso (costo vs protección).
- Restricciones por probabilidad (chance constraints): se exige que la probabilidad de violar la capacidad o ventana sea menor a α.
- *Simulación–optimización:* se optimiza para una solución candidata y se evalúa mediante simulación de Monte Carlo para estimar la confiabilidad y la variabilidad; luego se ajusta la búsqueda.

Estas opciones reflejan el giro actual hacia modelos que no solo "encuentran una ruta" sino que controlan el desempeño bajo incertidumbre (Ulmer, 2020; Soeffker *et al*., 2022; Adamo *et al*., 2024; Özarık *et al.,* 2021).

Para evitar ambigüedades (y mejorar la replicabilidad), conviene declarar explícitamente la estructura de la información: qué se decide antes de observar la incertidumbre y qué puede ajustarse durante la operación. En la práctica, esto puede comunicarse como un esquema de dos niveles: (i) un plan base (asignación y secuencia inicial) y (ii) acciones de ajuste permitidas cuando se revelan eventos (variación de tiempos, aparición o cancelación de órdenes, ausencia del cliente). Esta distinción es crucial porque dos modelos pueden llamarse "estocásticos" y, sin embargo, diferir radicalmente en lo que permiten cambiar durante el día; en consecuencia, no son comparables si no se explicita ese alcance. En términos de evaluación, esta estructura se traduce en medir no solo el costo esperado, sino también el impacto sobre el servicio y estabilidad del plan bajo escenarios, manteniendo control sobre el presupuesto computacional de cualquier ajuste (Ulmer, 2020; Soeffker *et al.,* 2022).

3.3. Función objetivo y extensiones modernas

La evaluación moderna de VRP ha mostrado que optimizar únicamente la distancia/costo puede producir soluciones frágiles ante la variabilidad (Bai *et al*., 2023). Por ello, se recomienda definir una función objetivo que combine eficiencia y estabilidad operativa.

Objetivo base (determinista). Una forma estándar:

$$min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^{k} + \sum_{i \in N} \sum_{k \in K} (\pi_e e_i^k + \pi_e l_i^k)$$

donde π_e π_l y ponderan la penalización por adelanto (si se modela como costo) y tardanza/violación (Xu *et al.,* 2020).

Extensión con confiabilidad/variabilidad (robustez). En presencia de incertidumbre, el costo esperado puede complementarse con un término de riesgo:

- Minimizar *costo esperado + variabilidad:*

$$minE[C] + \lambda Var(C)$$

- Minimizar **co***sto esperado + riesgo de cola:*

$$minE[C] + \lambda CVaR_{\alpha}(C)$$

Aunque este modelo no fija una sola métrica, sí establece el principio de que reportar el desempeño medio y la dispersión sea un requisito de transparencia experimental (Accorsi *et al.,* 2022; Bai *et al.,* 2023).

Asimismo, cuando existe un servicio diferenciado, la función objetivo debe reflejarlo de forma trazable. Una forma estándar es introducir penalizaciones ponderadas por prioridad p_i, de modo que la tardanza, el incumplimiento o la falla de entrega tengan impacto distinto según la criticidad del cliente o del acuerdo de servicio. Esto no implica "forzar" un único criterio, sino garantizar que el modelo represente decisiones reales: cuánto está dispuesto a pagar por sostener el servicio en el segmento prioritario y qué degradación se acepta en el resto. En ese sentido, la función objetivo funciona como un contrato explícito entre eficiencia y servicio, y su configuración debe ser consistente con las métricas reportadas en el protocolo experimental (Xu *et al*., 2020; Özarık *et al*., 2021; Accorsi *et al.,* 2022).

3.4. Instancias y evaluación comparativa: cómo construir bancos de prueba realistas (más allá de ajustes ad-hoc)

Uno de los problemas recurrentes en VRP y en enfoques ML para VRP es evaluar con instancias demasiado "limpias" o generadas ad-hoc que favorecen un método específico. Para evitarlo, el banco de pruebas debe construirse con reglas replicables, una distribución controlada y una diversidad estructural (Accorsi *et al.,* 2022; Bai *et al*., 2023).

Principios para instancias realistas:

- *Diversidad estructural:* densidad urbana vs rural, clústeres vs dispersión, distinto tamaño (n).
- *Heterogeneidad de restricciones:* capacidades distintas, ventanas estrictas y blandas, presencia incierta y horarios dependientes (Adamo et al., 2024; Özarık et al., 2021).
- *Escenarios repetibles:* misma semilla, misma distribución, mismos parámetros publicados.
- *Separación entrenamiento/prueba* (cuando hay ML): evitar "filtración" de información, y reportar desempeño fuera de distribución (Bengio et al., 2021; Accorsi et al., 2022).

Metodología sugerida de construcción (procedimiento).

- Paso A: Definir el contexto (última milla, distribución regional, urbano).
- Paso B: Generar nodos: distribución mixta (clústeres + ruido), o basada en polígonos/zonas (si es sintético).
- Paso C: Generar demandas y servicios: distribución (p. ej., discreta o lognormal) con posibles correlaciones entre clústeres.
- Paso D: Generar ventanas**:** intervalos (a_i, b_i) con distintos anchos, y parámetros de flexibilidad $\overline{\delta}_i^{\pm}$
- Paso E: Generar viaje:
  - determinista (distancia euclídea o matriz fija), y/o
  - dependiente del tiempo por franjas con multiplicadores (p_{ij}^h) (Adamo et al., 2024).
- Paso F: Generar incertidumbre: escenarios o perturbaciones de demanda/servicio/viaje con semilla y parámetros explícitos (Ulmer, 2020; Soeffker et al., 2022).
- Paso G: Versionado: publicar dataset, scripts de generación, seeds y metadatos.

Para que el banco de pruebas sea verdaderamente reutilizable, junto con las instancias conviene publicar un manifiesto técnico mínimo (metadatos): distribución espacial usada, parámetros de generación, rangos de demanda y servicio, regla de construcción de ventanas (anchos, tolerancias y flexibilidad), definición del perfil dependiente del tiempo (franjas, multiplicadores), y descripción de cómo se generan escenarios de incertidumbre (distribuciones, correlaciones y semilla). En enfoques con aprendizaje automático, además, se recomienda documentar explícitamente la partición entre conjuntos de ajuste/entrenamiento y conjuntos de prueba, y reportar resultados fuera de distribución cuando se aplique (Accorsi *et al.,* 2022; Bengio *et al.,* 2021). Esto evita que un método "gane" por el diseño del banco de pruebas y fortalece la transferibilidad de la evidencia a operaciones reales.

En la literatura reciente se promueve conectar la evaluación con entornos "tipo gemelo digital" para la logística urbana: no necesariamente como una implementación completa, sino como una filosofía de trazabilidad y consistencia entre datos, supuestos y evaluación (Belfadel *et al.,* 2023).

3.5. Protocolo de evaluación: métricas, comparación estadística, transparencias de cómputo

Evaluar VRP con rigor implica medir calidad de solución**,** robustez**,** tiempo, y asegurar comparabilidad. Esto es particularmente crítico cuando se usan ML, metaheurísticas y enfoques híbridos, porque el riesgo de "overfitting experimental" es alto (Accorsi *et al.,* 2022; Bai *et al.,* 2023; Bogyrbayeva *et al.,* 2024).

Métricas mínimas (recomendadas):

- *Costo total:* distancia/costo, penalizaciones por violación de la ventana, número de vehículos utilizados.
- *Nivel de servicio:* tardanzas promedio, tardanza máxima, % de clientes atendidos dentro de ventana base y dentro de ventana flexible.

- *Robustez* (si hay incertidumbre): media y dispersión (desviación estándar, percentiles), o tasa de factibilidad bajo escenarios (Ulmer, 2020; Soeffker *et al*., 2022).
- *Eficiencia computacional:* tiempo por instancia, presupuesto de evaluaciones, y escalabilidad por tamaño (n).
- *Huella de cómputo*: CPU/GPU, memoria, versiones, número de hilos, compilador y sistema operativo.

Diseño experimental reproducible. Se recomienda:

- Múltiples semillas para métodos estocásticos (mínimo 10–30 por instancia, según el prosupuesto).
  comparable (tiempo máximo o número de evaluaciones).
- *Mismo conjunto de instancias* y mismas reglas de preprocesamiento.
- *Reporte de hiperparámetros* completos (y cómo se obtuvieron: manual vs automático).
- Publicación de logs: semillas, rutas, costos, métricas por corrida (Accorsi *et al*., 2022).

Compa
. En lugar de “ganó en promedio”, se recomienda reportar:

- Pruebas pareadas (p. ej., Wilcoxon signed-rank) cuando se comparan dos métodos bajo mismas instancias/semillas,
- Análisis de ranking cuando hay múltiples algoritmos (reportar distribución de ranks y medidas robustas).

Lo esencial es que la comparación sea justa, transparente y replicable, especialmente en ML aplicado a optimización combinatoria (Accorsi et al., 2022; Bengio et al., 2021; Bogyrbayeva et al., 2024).

Para reforzar la interpretación de resultados, se recomienda complementar la significancia con magnit
ud del efecto e intervalos de confianza. En particular, además del p-valor, reportar diferencias medianas por instancia, medidas robustas (p. ej., medianas y percentiles) y un indicador de tamaño de efecto permite distinguir mejoras marginales de mejoras operativamente relevantes. Cuando se comparan múltiples métodos, conviene declarar la estrategia de control de comparaciones (por ejemplo, análisis por ranking o ajustes en el criterio de decisión) para evitar conclusiones infladas por “selección” posterior. Finalmente, la transparencia computacional debe cerrarse con una ficha reproducible del entorno (CPU/GPU, RAM, número de hilos, sistema operativo, compilador/librerías, y configuración de aleatoriedad), de modo que terceros puedan repetir el experimento de forma razonable (Accorsi *et al*., 2022; Bai *et al*., 2023; Bogyrbayeva *et al*., 2024).

3.6. Buenas prácticas contemporáneas para el ajuste de parámetros (automatización, robustez y reporte reproducible)

El ajuste de parámetros no debe ser un “arte oculto”. En metaheurísticas y ALNS, el rendimiento depende fuertemente del tuning; por ello, el reporte debe indicar: qué se ajustó, con qué presupuesto, sobre qué instancias y cómo se evitó el sobreajuste (Mara *et al*., 2022; Voigt, 2025).

Buenas prácticas recomendadas:

- Automatizar el tuning: usar procedimientos sistemáticos (búsqueda aleatoria, optimización bayesiana, racing) en lugar de ajustes manuales sin trazabilidad.
- Separar el tuning y la prueba: el conjunto de instancias para ajustar no debe coincidir con el de reporte final, especialmente si el método usa ML o aprendizaje de operadores (Accorsi et al., 2022; Bai et al., 2023).

- ajustar para buen desempeño promedio *y* estabilidad; reportar sensibilidad: “qué pasa si muevo ±10% este parámetro”.
- *Presupuesto explícito:* número de evaluaciones, tiempo máximo, semillas usadas.
- **Reporte mínimo**: valores finales, rangos explorados, criterio de selección (mejor media, mejor percentil, mejor CVaR), y logs publicables.

ALNS y operadores adaptativos. Dado el peso actual de ALNS, es recomendable:

- Explicitar el conjunto de operadores de destrucción/reparación,
- Indicar cómo se actualizan pesos (adaptación),
- Reportar qué operadores dominan en distintos escenarios.

La literatura reciente ha avanzado en taxonomías, aplicaciones y análisis de operadores en ALNS, lo que refuerza que “comparar ALNS” exige transparencia sobre los componentes internos (Mara *et al.,* 2022; Voigt, 2025).

Conexión con ML y enfoques híbridos. Cuando se integra ML (p. ej., seleccionar operadores, estimar costos, guiar construcción), se debe reportar:

- Arquitectura y entrenamiento (si aplica),
- Validación, split y generalización,
- Comparación con líneas base sólidas (heurísticas y exactas) bajo el mismo presupuesto.

Esto es coherente con guías específicas para el testing computacional de ML en VRP (Accorsi *et al., 2022) y con el marco metodológico general de ML para optimización combinatoria (Bengio et* al., 2021; Bogyrbayeva *et al.,* 2024).

Como práctica de cierre, el ajuste de parámetros puede implementarse en tres pasos simples y auditables: (i) definir un conjunto de instancias de ajuste (separado del de prueba) y un presupuesto explícito de tuning; (ii) ejecutar un procedimiento sistemático (búsqueda aleatoria, optimización bayesiana o racing) con múltiples semillas por configuración, registrando resultados completos; y (iii) seleccionar parámetros por un criterio que combine desempeño medio y estabilidad (por ejemplo, media + dispersión o percentiles), reportando además sensibilidad local para identificar parámetros críticos. Con ello, el tuning deja de ser una decisión opaca y se convierte en un proceso reproducible que reduce el riesgo de sobreajuste experimental, especialmente cuando el método incluye componentes adaptativos o basados en aprendizaje (Accorsi et al., 2022; Mara *et al*., 2022; Voigt, 2025).

3.7. Protocolo de ejecución reproducible y transferencia a operación

En problemas de ruteo bajo incertidumbre, la calidad de un aporte no depende únicamente del “mejor costo” reportado, sino de la capacidad de sostener comparaciones justas, explicar decisiones de diseño y permitir que terceros repliquen—o, al menos, verifiquen—los resultados con un grado razonable de fidelidad. Este principio es particularmente relevante cuando se emplean metaheurísticas y procedimientos estocásticos (p. ej., ALNS, VNS, búsquedas híbridas), donde pequeñas variaciones en semillas, penalizaciones o reglas de aceptación pueden alterar el desempeño observado y, por consiguiente, sesgar conclusiones sobre robustez y generalización. En este contexto, se propone formalizar un protocolo de reporte mínimo y un **contrato de ejecución**, entendidos como un conjunto de decisiones “congeladas” que definen la identidad operativa del método y hacen posible su evaluación transparente (Martín-Santamaría *et al*., 2024).

3.7.1. Notas de reporte mínimo para comparación justa

El reporte mínimo cumple un rol análogo al de un "instrumento de medición": si no se define con precisión, las comparaciones quedan contaminadas por diferencias ocultas de implementación. El objetivo es que cualquier lector pueda responder, sin ambigüedad, qué se optimizó**,** cómo se definió la factibilidad **y** qué decisiones se dejaron al azar**.**

(i) Política de puntaje (score) y prioridades.

Debe declararse explícitamente la regla de priorización usada para construir solución inicial o para despachar en línea (por ejemplo: EDF, slack, o puntaje ponderado). Se recomienda publicar la ecuación de score y el significado operacional de cada término (urgencia temporal, criticidad de demanda, confiabilidad del servicio, etc.). Cuando el score incluye normalizaciones, debe especificarse el rango y el método de escalado para evitar sesgos por magnitud.

(ii) Pesos y penalizaciones.

En enfoques con restricciones blandas (SOFT), las penalizaciones son, de facto, parte del modelo: cambian la geometría del paisaje de búsqueda. Por ello, deben reportarse (a) las penalizaciones por violación (capacidad, ventanas de tiempo, continuidad, etc.), (b) cualquier esquema de penalización adaptativa, y (c) si los costos incluyen componentes fijos (activación de vehículo), variables (distancia/tiempo), o de servicio (tardanza, rechazos). Este punto es crítico para no confundir "mejora de costo" con "relajación encubierta" de factibilidad.

(iii) Factibilidad HARD vs SOFT.

Debe indicarse qué restricciones son **duras** (inviolables) y cuáles son **blandas** (permitidas con penalización). En operación real, esta distinción representa compromisos gerenciales: por ejemplo, capacidad y precedencias suelen ser HARD, mientras que ciertos márgenes de ventana pueden modelarse como SOFT bajo acuerdos de servicio. En estudios comparativos, la clasificación HARD/SOFT debe mantenerse constante entre métodos para no comparar soluciones de naturalezas distintas.

(iv) Reglas de desempate.

Los empates "aparentemente inocentes" (igual score, igual costo incremental) generan divergencias en rutas finales. Se recomienda declarar reglas de desempate en cascada: p. ej., (1) menor slack, (2) menor incremento de tardanza, (3) menor desvío, (4) menor índice/ID. Si existe aleatoriedad en desempates, debe registrarse como parte del componente estocástico.

(v) Tratamiento de espera.

La espera puede modelarse como (a) tiempo permitido sin costo, (b) costo de espera, (c) restricción operacional (p. ej., no esperar más de X), o (d) decisión de reubicación (waiting en nodos "estratégicos"). En ruteo con incertidumbre, esta decisión se conecta con políticas de anticipación y robustez (p. ej., esperar para reducir probabilidad de tardanza vs reencaminar para aumentar cobertura). Debe precisarse si la espera se permite en cliente, en depósito o en puntos de espera.

(vi) Semillas y aleatoriedad.

Si el método es estocástico, debe informarse: (a) si hay aleatoriedad en selección de operadores, (b) si la construcción inicial incluye aleatoriedad, (c) el número de semillas y cómo se generan, y (d) si se fija la semilla por corrida o por instancia. Sin este dato, la replicación es impracticable y la varianza del método queda invisible (Martín-Santamaría et al., 2024).

(vii) Presupuesto de cómputo como baseline.

Cuando se comparan heurísticas, el presupuesto computacional (tiempo, iteraciones, evaluaciones) debe tratarse como un recurso explícito. Reportar solo "mejor valor" sin controlar presupuesto favorece métodos más costosos en cómputo. Se recomienda fijar: (a) tiempo máximo por instancia, (b) criterio de parada, (c) número de corridas, y (d) hardware/entorno (CPU, RAM, lenguaje, librerías relevantes) como contexto mínimo.

Este ciclo también clarifica el rol de tecnologías como gemelos digitales: su aporte no es "decorativo", sino estructural, porque ordenan el proceso completo (captura → simulación/optimización → ejecución → monitoreo), reduciendo la brecha entre formulación y operación.

3.7.2. El "contrato de ejecución": qué se debe congelar

Además de reportar, es útil congelar decisiones para que el método sea auditable y transferible. El contrato de ejecución define el conjunto de componentes que deben permanecer invariantes durante un experimento (o durante una etapa operacional), salvo que se declare una modificación controlada.

Congelamientos para ALNS u otras metaheurísticas con operadores

Para una arquitectura tipo ALNS, se recomienda congelar:

1. *Lista exacta de operadores* (destroy y repair), incluyendo variantes parametrizadas (p. ej., "worst removal con k%", "related removal con métrica de similitud", "greedy insertion vs regret-k").
2. *Distribución inicial* o probabilidad inicial de selección de operadores.
3. *Regla de aceptación*, especificando si se usa SA-like, record-to-record, threshold, o aceptación determinista; y, cuando aplique, parámetros como T0T_0T0 y α\alphaα (enfriamiento).
4. *Regla de actualización de pesos* (p. ej., mecanismo tipo "bandit"), con parámetros ρ\rhoρ, LLL, σ\sigmaσ (tasa de olvido, longitud de segmento, recompensas).
5. *Criterio de parada* (tiempo, iteraciones sin mejora, presupuesto de evaluaciones).
6. Número de corridas/semillas por instancia, con el esquema de agregación (promedio, mediana, mejor de N).

Este congelamiento es coherente con la filosofía original de ALNS: operadores compiten y su frecuencia se ajusta por desempeño histórico, pero esa competencia debe estar definida de manera trazable para evitar "ALNS a la carta" (Røpke & Pisinger, 2006).

Con este mínimo, es posible sostener análisis de robustez y confiabilidad, así como justificar el costo total cuando incorpora componentes que suelen omitirse en modelos simplificados (espera, horas extra y penalizaciones).

3.7.3. Congelamientos para horizonte rodante y reoptimización

En operación dinámica, el método no solo produce una ruta: produce una política de actualización**.** Por tanto, deben congelarse:

- *Triggers de reoptimización:* por evento (nuevo pedido, cancelación, atraso, incidente) y/o por Δt\Delta tΔt (cada X minutos).
- *Ventana de congelamiento (freeze):* qué parte del plan queda "bloqueada" para proteger estabilidad operacional (p. ej., próximas 20–40 min o próximas 2 paradas).
- *Dos niveles de cómputo:* un modo "rápido" para reacción inmediata y un modo "profundo" para reoptimización cuando el sistema lo permite (p. ej., baja carga o fuera de picos).
- *Criterio de aceptación de cambios:* cuánto desvío del plan se permite antes de ejecutar una reoptimización, y cómo se penaliza la inestabilidad (p. ej., costo por "cambio de secuencia" o "cambio de asignación").

Estos elementos son centrales en esquemas de rolling horizon aplicados a ruteo y logística dinámica: el rendimiento no depende solo de resolver bien un subproblema, sino de **cómo** se encadenan decisiones en el tiempo (Bjelić *et al*., 2022; Keskin e*t al*., 2023).

3.7.4. Contrato de ejecución: comparabilidad y replicabilidad como condiciones de credibilidad

En algoritmos modernos, y especialmente en ALNS, el desempeño depende fuertemente de decisiones "pequeñas" (operadores, aceptación, penalizaciones, cálculo de espera, reglas de desempate). Por ello, para sostener comparaciones defendibles, es necesario declarar un *contrato de ejecución:* un conjunto de componentes que se consideran congelados para que el experimento sea replicable y para que el lector pueda atribuir diferencias al método y no a variaciones ocultas de implementación.

En ALNS, el contrato mínimo debe fijar:
Lista exacta de operadores destroy/repair**,** incluyendo distribución inicial de selección (pesos iniciales) y cualquier heurística adicional (p. ej., inserción regret-k o "time-window insertion").

- *Regla de aceptación:* determinista o tipo SA-like; si es SA-like, declarar la temperatura inicial T_0 y el esquema de enfriamiento (por ejemplo, $T \leftarrow \alpha T$).
- *Regla de actualización de pesos:* parámetros y ventana de aprendizaje (p. ej., ρ, L, σ) y cómo se puntúa cada operador (mejora, aceptación, diversidad).
- *Criterio de parada:* tiempo, iteraciones o evaluaciones, y su relación con el presupuesto computacional.
- Número de corridas y semillas**:** cuántas replicaciones por instancia, cómo se fijan semillas y cómo se reportan resultados (promedio, mejor, dispersión).

Este contrato es igualmente aplicable a procedimientos de referencia: no se trata de "cualquier heurística rápida", sino de un esquema comparativo con reglas y presupuesto claramente definidos.

3.7.5. De la instancia "de papel" a los datos operativos

Una brecha recurrente entre investigación y empresa radica en la diferencia entre instancias limpias (coordenadas, demandas, tiempos fijos) y operación real (eventos, tráfico, latencias de información, restricciones tácitas). Para cerrar esa brecha, se recomienda explicitar dos capas de datos:

Capa mínima (para resolver):

- Geometría (clientes, depósitos, zonas), demandas y capacidades.
- Ventanas de tiempo (duras o blandas) y tiempos de servicio.
- Flota (tipos, costos fijos/variables, restricciones por zona).
- Matriz de tiempos/distancias o acceso a un motor de ruteo.

Capa de instrumentación operativa (para aprender y gobernar):

- Trazas por vehículo: tiempos reales, paradas, desviaciones, esperas.
- Eventos: pedidos tardíos, cancelaciones, rechazos, incidencias.
- Indicadores de calidad: cumplimiento, puntualidad, estabilidad del plan.
- Versionado: modelo, parámetros, semillas, y configuración de ejecución.

La segunda capa no es "decorativa": permite auditoría, análisis post-mortem y mejora continua. En ruteo dependiente del tiempo y reoptimización en línea, la calidad del sistema depende de incorporar información contextual (tráfico, red vial, predicción) para evitar que el plan sea óptimo solo bajo un supuesto estático (Adamo *et al*., 2024; Gmira *et al*., 2021).

3.7.6. Operación dinámica: estabilidad, calidad de servicio y "costo del cambio"

Cuando un plan se recalcula con frecuencia, emerge un costo que los modelos clásicos tienden a omitir: el costo del cambio. Cambiar asignaciones o secuencias puede generar fricción (comunicación al conductor, reconfiguración, promesas al cliente), y esa fricción afecta la

percepción de confiabilidad. Por ello, se sugiere tratar la estabilidad como un objetivo explícito, por ejemplo:

- Penalizar cambios de ruta dentro de la ventana congelada.
- Minimizar número de reasignaciones de pedidos ya "comprometidos".
- Limitar cambios de secuencia salvo que se supere un umbral de riesgo (tardanza esperada, congestión, ruptura de capacidad).

Esta visión es coherente con la literatura de ruteo dinámico: el problema no es solo "optimizar", sino hacerlo con control de perturbaciones y gobernanza del proceso decisional, donde el horizonte rodante funciona como un mecanismo para equilibrar reacción y consistencia (Bjelić *et al*., 2022; Keskin *et al.,* 2023).

3.7.7. Gemelos digitales como puente: evaluación continua y transferencia

Una línea particularmente fértil para consolidar transferencia a operación consiste en integrar el ruteo con un gemelo digital logístico: una representación computacional del sistema que sincroniza datos (estado de flota, eventos, red) y permite simular escenarios, probar políticas y anticipar fallas. En este marco, el gemelo digital no reemplaza al optimizador; lo contextualiza y lo somete a evaluación continua: ¿cómo se comporta la política bajo congestión severa, bajo cambios de demanda, bajo fallos de vehículos, bajo restricciones emergentes?

La literatura reciente sintetiza este potencial y también sus retos: interoperabilidad de datos, confiabilidad de modelos, y necesidad de transparencia para que los tomadores de decisión confíen en recomendaciones algorítmicas (Le & Fan, 2024).

En términos prácticos, el gemelo digital habilita tres capacidades que el libro busca enfatizar:

1. *Evaluación realista:* comparar políticas bajo distribuciones de incertidumbre plausibles, más allá de instancias estáticas.
2. *Diagnóstico operacional:* detectar si incumplimientos provienen de modelado (supuestos), de ejecución (operación), o de datos (latencia/ruido).
3. *Mejora iterativa gobernada:* ajustar parámetros y reglas con trazabilidad, evitando "tuning" opaco.

3.7.8. Transparencia experimental: registro, versionado y trazabilidad

En línea con una metodología de experimentación reproducible, se recomienda complementar los resultados con un esquema explícito de trazabilidad experimental, de modo que cada corrida pueda ser auditada y replicada sin ambigüedades. Esto implica registrar, como mínimo, (i) versión del código y dependencias, (ii) configuración exacta de parámetros y penalizaciones, (iii) semillas utilizadas y número de replicaciones por instancia, (iv) presupuesto computacional asignado, y (v) métricas reportadas (incluyendo dispersión y percentiles cuando aplique). La adopción sistemática de este registro no solo mejora la calidad metodológica del estudio, sino que reduce el riesgo de conclusiones espurias derivadas de configuraciones no controladas, fortaleciendo la comparabilidad entre métodos y la credibilidad de los hallazgos (Martín-Santamaría et al., 2024).

Capítulo 4. Métodos de solución: de metaheurísticas clásicas a enfoques híbridos y optimización asistida por aprendizaje

4.1. Línea base: heurísticas constructivas y prioridades por ventanas

Las soluciones base cumplen dos funciones: (i) proveer un punto de comparación estable y reproducible, y (ii) servir como generadores de soluciones iniciales para metaheurísticas y ALNS. En literatura reciente se insiste en que, especialmente al evaluar enfoques con ML, los métodos de referencia deben ser sólidos y bien documentados para evitar comparaciones sesgadas (Accorsi *et al.,* 2022; Bai *et al.,* 2023).

En problemas con ventanas de tiempo, las heurísticas constructivas pueden reinterpretarse como "políticas de despacho": reglas que determinan a qué cliente servir a continuación, cuándo esperar, y cómo cerrar o abrir nuevas rutas según el estado del sistema (tiempo, capacidad, holguras y proximidad). Esta perspectiva conecta naturalmente con escenarios dinámicos y con sistemas de ajuste dinámico de rutas en tiempo real (Ulmer, 2020; Soeffker, Ulmer, & Mattfeld, 2022).

Para que estas soluciones base cumplan su papel metodológico (comparación justa y reutilización como punto de partida), no basta con nombrarlas: es necesario fijar explícitamente (a) la estructura de datos usada, (b) la verificación incremental de factibilidad (capacidad y ventanas), (c) la forma de calcular el costo incremental (incluyendo esperas y penalizaciones si hay ventanas flexibles) y (d) reglas deterministas de desempate. De lo contrario, dos implementaciones "aparentemente iguales" pueden producir diferencias no atribuibles al método, sino a detalles de programación, afectando la validez de la comparación (Accorsi *et al*., 2022; Bai *et al*., 2023).

En el contexto del modelo (tiempos dependientes del tiempo y ventanas flexibles), la "factibilidad" debe evaluarse con el mismo criterio en todas las líneas base. Por ejemplo: si el tiempo de viaje depende de la hora, el chequeo temporal debe usar el tiempo de salida efectivo al recorrer cada arco; y si las ventanas son flexibles, el método debe decidir explícitamente si (i) prohíbe las violaciones (modo estricto) o (ii) permite las violaciones con penalización (modo flexible). Esta distinción es clave porque cambia el carácter de la solución inicial: en modo flexible, la solución tiende a ser menos "frágil" ante incertidumbre, y además es coherente con una evaluación posterior basada en costo total y servicio (Ulmer, 2020; Soeffker *et al*., 2022).

4.1.1. Heurísticas constructivas clásicas (adaptadas a ventanas)

Se recomiendan al menos tres familias de constructivos como línea base, implementadas con el mismo estilo de datos y verificación de factibilidad:

- Inserción secuencial por cercanía (Nearest Neighbor con chequeo de factibilidad): elige el siguiente cliente factible de menor costo incremental.
- Inserción más barata (Cheapest Insertion): inserta al cliente no atendido cuya inserción en la posición más factible minimiza el costo incremental total.
- Ahorros (Savings) con factibilidad: combina rutas parciales siempre que la unión respete la capacidad y las ventanas; útil como baseline rápido.

4.1.2. Prioridades por ventanas: EDF y variantes como políticas

En presencia de ventanas de tiempo, una regla robusta y fácil de interpretar consiste en priorizar según la "urgencia temporal". La forma más conocida es EDF (Earliest Due Date), donde se selecciona el cliente con menor b_i (fin de ventana) entre los candidatos factibles. La lógica es simple: si un cliente "vence" antes, se atiende primero para reducir el riesgo de tardanza o de inviabilidad. Sin embargo, el EDF puro puede generar decisiones miopes al ignorar el componente espacial (distancia/costo incremental), por ejemplo, eligiendo un cliente "urgente" pero muy lejano, lo que termina afectando la factibilidad y el costo total de la ruta. Por ello, en aplicaciones reales se recomienda usar prioridades híbridas (weighted priorities) que equilibren "tiempo" y "espacio", actualizadas dinámicamente según el estado de la ruta.

Ejemplos de prioridades (recomendación: definirlas y reportarlas explícitamente):

- *Urgencia por fecha límite (EDF):* ordenar los candidatos por b_i ascendente.
- Urgencia por holgura (slack-based): priorizar por la holgura temporal estimada, por ejemplo:

$$slack(i) = b_i - \hat{t}_{llegada}(i)$$

Donde los valores más pequeños indican un mayor riesgo y, por tanto, una mayor prioridad.

- *Cercanía / costo incremental:* priorizar por el costo de inserción local Δc(i) (o por el mejor incremento total al insertar en alguna posición factible).
- *Compromiso espacio–tiempo (score ponderado):* definir un puntaje que combine los factores anteriores, por ejemplo:

$$score(i) = w_1 \cdot \Delta c(i) + w_2 \cdot penalidad_urgencia(i) + w_3 \cdot penalidad_riesgo(i)$$

Y seleccionar al cliente con el score mínimo, fijando reglas de desempate deterministas.
En variantes con ventanas blandas (soft time windows), estas políticas deben incorporar explícitamente el costo de la tardanza y/o de las holguras. En lugar de "bloquear" la inserción cuando se viola la ventana, el sistema puede permitirla con una penalización, lo que hace que la solución inicial sea más estable (menos frágil) y más cercana a la operación real, especialmente cuando existen incertidumbres en el viaje o el servicio. En este sentido, las prioridades por ventanas no se entienden solo como una heurística, sino como una política de despacho: una regla operativa que decide a quién atender después en función de la urgencia, el costo y el riesgo, y que puede integrarse naturalmente con ajustes dinámicos de rutas en tiempo real y marcos híbridos de mejora.

Esta lectura de "política de despacho" es especialmente importante cuando el problema evoluciona en el tiempo: no se trata únicamente de construir una buena ruta, sino de definir una regla coherente de toma de decisiones locales que pueda reactivarse cuando cambian las condiciones (nuevas órdenes, congestión, fallas). Por eso, para fines de evaluación científica, estas prioridades deben reportarse como parte del método (y no como detalle implícito), de modo que su contribución pueda auditarse y replicarse (Ulmer, 2020; Soeffker *et al*., 2022; Accorsi *et al.,* 2022).

4.1.3. Plantilla mínima reproducible para un método de referencia (pseudocódigo)

Una buena práctica es fijar una plantilla única para todos los métodos de referencia (misma estructura de datos, mismos chequeos, mismas reglas de desempate). A continuación, se propone una plantilla de despacho reproducible:

```
POLITICA_DESPACHO_SOLUCION_INICIAL(instancia, pesos w):
  rutas = []
  clientes_pendientes = N
  mientras clientes_pendientes no vacío:
    iniciar nueva ruta r desde depósito
    mientras exista cliente factible para insertar en r:
      para cada cliente i en clientes_pendientes:
```

```
        calcular score(i | estado_ruta, w)
      elegir i* con score mínimo (y desempates deterministas)
      insertar i* en la mejor posición factible de r
      remover i* de clientes_pendientes
    cerrar r y añadir a rutas
  retornar rutas
```

Junto con esta plantilla, se recomienda declarar explícitamente los elementos mínimos de reporte para asegurar comparaciones justas: (i) el criterio de priorización o puntaje utilizado (por ejemplo, fecha límite/EDF, holgura, costo incremental o combinación ponderada); (ii) los pesos y penalizaciones empleados; (iii) el criterio de factibilidad adoptado (estricto o permitido con penalización); (iv) las reglas de desempate, de forma determinista; (v) el tratamiento de la espera (si se permite y cómo se contabiliza); (vi) cualquier aleatoriedad incorporada (semillas, generador, número de replicaciones); y (vii) el presupuesto computacional asignado cuando este método se use como comparador. Este nivel de detalle evita diferencias espurias entre implementaciones y fortalece la comparabilidad y la reproducibilidad, especialmente al contrastar enfoques que incorporan componentes de aprendizaje automático (Accorsi *et al*., 2022; Bai *et al*., 2023).

4.2. Motores metaheurísticos comparables

Para construir comparaciones justas, conviene describir metaheurísticas por "familias" (mecanismo de exploración) y por "operadores" reutilizables (vecindarios, cruces, mutaciones, reparaciones), evitando implementaciones ad-hoc difíciles de replicar. Esta filosofía se alinea con guías modernas de evaluación y prácticas de reporte para VRP y ML-VRP (Accorsi *et al*., 2022; Bogyrbayeva *et al*., 2024).

En este marco, "motor comparable" significa que diferentes métodos se ejecutan bajo un contrato de evaluación común, de tal forma que las mejoras observadas puedan atribuirse al mecanismo metaheurístico y no a ventajas ocultas de implementación. En VRP con restricciones ricas (ventanas flexibles, tiempos dependientes del tiempo, prioridades), la comparabilidad depende especialmente de que todos los métodos compartan la misma lógica: (i) verificación incremental de factibilidad (capacidad y calendarios), (ii) cálculo del costo total (incluyendo espera y penalizaciones), y (iii) tratamiento de la variación (semillas, repetición y presupuesto computacional). Este enfoque está alineado con guías recientes de evaluación reproducible y con revisiones sobre ML aplicado a optimización combinatoria, donde el principal riesgo no es "equivocarse de algoritmo", sino comparar bajo condiciones no equivalentes (Accorsi *et al*., 2022; Bogyrbayeva *et al*., 2024).

Por ello, al describir cada familia de metaheurísticas, se recomienda reportar un mínimo homogéneo: (a) representación de la solución (rutas explícitas o "secuencia global" más partición), (b) biblioteca de movimientos/operadores (vecindarios o ediciones), (c) función de costo con penalizaciones y reglas de espera, (d) mecanismo de aceptación (si se permiten empeoramientos y bajo qué regla), (e) criterio de parada (tiempo/iteraciones/evaluaciones) y (f) control de aleatoriedad (semillas, número de corridas por instancia). Esta estandarización facilita comparaciones defendibles y evita conclusiones infladas por configuraciones favorables o por asimetrías presupuestarias (Accorsi *et al*., 2022; Bai *et al*., 2023).

4.2.1. Métodos de trayectoria: Simulated Annealing (SA) y Tabu Search (TS)

Los métodos de trayectoria exploran el espacio de soluciones avanzando desde una solución actual hacia vecinas, controlando diversificación y evitando estancamiento. SA introduce una probabilidad de aceptar empeoramientos controlada por una "temperatura"; TS prohíbe volver

a movimientos recientes mediante una lista tabu (Kirkpatrick, Gelatt, & Vecchi, 1983; Glover, 1986).

En problemas con ventanas de tiempo y penalizaciones por servicio, el desempeño de SA/TS depende menos del "nombre del método" y más de cómo se definen tres piezas: (i) la estructura de vecindarios (qué cambios se permiten), (ii) la evaluación incremental (cómo se recalculan tiempos y penalizaciones sin recomputar toda la ruta) y (iii) el control del presupuesto (cuántos movimientos se evalúan por unidad de tiempo). Por ejemplo, en ruteo dependiente del tiempo, un movimiento local no solo cambia la distancia, sino el calendario de visitas; por lo tanto, la evaluación debe actualizar coherentemente las llegadas/salidas y recalcular las tardanzas/esperas antes de aceptar el cambio. Esta disciplina de evaluación es crucial, ya que luego se compararán familias distintas bajo un protocolo único (Accorsi *et al*., 2022).

Componentes comparables recomendados:

- *Representación:* ruta-secuencia (giant tour) + partición, o conjunto de rutas explícitas.
- Vecindarios: 2-opt, Or-opt, relocate, swap, 2-opt* entre rutas (en ventanas, con verificación incremental).
- *Función de costo:* costo espacial + penalizaciones por tardanza/holguras + penalización por sobrecapacidad (si se permite la infeasibilidad temporal).
- *Criterio de parada:* presupuesto de tiempo o de iteraciones; múltiples semillas para mayor robustez.

En VRP con ventanas, una estrategia robusta es permitir soluciones temporalmente infactibles con penalización y usar reparaciones periódicas, lo que mejora la exploración sin perder el control de la factibilidad. Este principio reaparece de forma sistemática en ALNS, donde la búsqueda puede operar sobre soluciones factibles o infactibles mediante una función de costo generalizada que suma costos operativos y penalizaciones proporcionales al grado de violación de las restricciones (Voigt, 2025; Mara et al., 2022).

4.2.2. Métodos poblacionales: Genetic Algorithms (GA)

Los GA operan sobre una población de soluciones, aplicando selección, cruce y mutación. El reto central en VRP es definir un *encoding* que preserve la estructura de rutas y permita operadores de cruce que produzcan descendientes razonables. En la práctica, se recomiendan representaciones del tipo "giant tour" con un procedimiento de "split" factible (Holland, 1975). En términos prácticos, la "codificación" debe permitir que los operadores de cruce y mutación conserven la estructura logística (subrutas coherentes) y no produzcan sistemáticamente soluciones inviables. Por ello, una práctica extendida es separar dos niveles: (a) una permutación global de clientes (secuencia) y (b) un procedimiento de partición ("split") que construye rutas factibles (o casi factibles) bajo capacidad y ventanas. Este desacoplamiento mejora la comparabilidad porque permite distinguir qué parte del desempeño proviene del operador evolutivo (cruce/mutación) y qué parte proviene del procedimiento de decodificación/partición, que también debe reportarse y mantenerse fijo entre variantes (Accorsi *et al*., 2022; Bai *et al*., 2023).

Operadores típicos y comparables:

- *Cruce por orden (OX)* o basado en segmentos, aplicado sobre la permutación de clientes (giant tour).
- *Mutaciones simples* (swap, relocate) y mutaciones estructurales (segment reverse).
- *Decodificación:* algoritmo *split* que genera rutas factibles (o casi factibles) bajo ventanas y capacidad.
- Estrategias elitistas y control de diversidad (evitar convergencia prematura).

4.2.3. Enjambres: Particle Swarm Optimization (PSO) y adaptaciones discretas

PSO fue diseñado para espacios continuos, pero existen adaptaciones discretas para problemas combinatorios, en los que la "velocidad" y la "posición" se interpretan como operadores o probabilidades de aplicar movimientos. Una visión útil es tratar el estado del enjambre como un conjunto de políticas de edición (mutaciones) guiadas por mejores historiales (Kennedy & Eberhart, 1995).

En la literatura reciente, las variantes discretas de PSO para ruteo suelen integrar mecanismos de mejora por vecindarios para reforzar la intensificación sin perder diversidad. En esta línea, se han propuesto enfoques en los que las "partículas" generan nuevas soluciones mediante operadores de edición (por ejemplo, extracción–reinserción, intercambio y reordenamientos) y, posteriormente, aplican una búsqueda local breve para consolidar mejoras, guiadas por soluciones élite y memorias históricas del enjambre. Este tipo de diseño ha mostrado resultados competitivos en VRP con ventanas de tiempo cuando se evalúa en bancos de prueba estándar y con métricas comparables (Wu et al., 2024).

Recomendaciones para un PSO discreto comparable en VRP:

- Definir una biblioteca fija de movimientos (swap/relocate/2-opt/2-opt*).
- Definir la actualización como una mezcla de: inercia (mantener el estilo), atracción a pbest y gbest (aplicar subconjuntos de movimientos).
- Restringir el presupuesto por partícula y reportar claramente el número de partículas y de iteraciones.

Para mantener la comparabilidad, en PSO discreto conviene explicitar dos elementos adicionales: primero, el mecanismo de factibilidad (cómo se reparan violaciones de capacidad/ventanas después de aplicar operadores), y segundo, el presupuesto interno por iteración, ya que una partícula que aplica muchos movimientos y además ejecuta una búsqueda local extensa no es comparable con una que solo aplica un movimiento simple. En consecuencia, al reportar PSO discreto se recomienda incluir: número de partículas, iteraciones, número máximo de movimientos por actualización, longitud de búsqueda local (si existe), y criterio de selección de soluciones élite (qué se considera pbest/gbest y con qué desempate). Este nivel de detalle permite auditar y replicar los resultados sin ambigüedades (Accorsi *et al.,* 2022; Bogyrbayeva *et al*., 2024).

4.2.4. Bio-inspiradas y el enfoque "Ébola" como antecedente histórico del caso

Las metaheurísticas bio-inspiradas incluyen familias basadas en selección natural, colonias, comportamiento animal y modelos epidemiológicos. En particular, existe una línea de algoritmos inspirados en la propagación de enfermedades, en la que los estados poblacionales guían la exploración y la explotación. Como antecedente histórico relacionado con el caso, se reporta el Ebola Optimization Search Algorithm (EOSA), propuesto a partir de un modelo de propagación de la enfermedad, y estudiado en *benchmarks* de optimización (Oyelade & Ezugwu, 2021; Oyelade & Ezugwu, 2022).

Para mantener el rigor, se recomienda usar estos enfoques como inspiración o como baseline secundario, evitando afirmar superioridad sin un protocolo de comparación sólido (Accorsi *et al.,* 2022). Si se implementan, deben documentarse: estados del modelo, parámetros epidemiológicos, tamaño poblacional, y presupuesto computacional; y compararse con baselines estándar (SA/TS/GA/PSO) bajo las mismas condiciones.

Metodológicamente, este tipo de enfoques se recomienda como comparador secundario o como inspiración para mecanismos de exploración, evitando sostener una superioridad sin un

protocolo estricto de comparación. Cuando se incluyan, deben documentarse con precisión: tamaño poblacional, parámetros del modelo inspirado (tasas y transiciones), criterio de parada y presupuesto computacional, y compararse bajo las mismas reglas de evaluación y reporte que el resto de las metaheurísticas del estudio (Accorsi *et al*., 2022).

Para cerrar esta sección y dar paso al marco ALNS, conviene subrayar que las familias revisadas (trayectoria, poblacionales y bio-inspiradas) comparten una dependencia crítica: el desempeño real proviene de la combinación entre representación, operadores y reglas de aceptación bajo un presupuesto computacional controlado. En problemas "rich", esa necesidad de modularidad y de control fino sobre los componentes (qué se modifica, cómo se repara, cómo se adapta la búsqueda) ha llevado a que la literatura privilegie arquitecturas que integran sistemáticamente destrucción y reconstrucción, con adaptación de operadores. En esa línea, el siguiente apartado presenta ALNS como un marco de referencia contemporáneo, precisamente por su capacidad de incorporar restricciones y criterios de servicio sin sacrificar la trazabilidad ni la comparabilidad experimental (Mara *et al*., 2022; Voigt, 2025).

4.3. Marco ALNS moderno (destroy/repair, pesos adaptativos, aceptación, intensificación/diversificación)

En la literatura reciente, ALNS se consolida no solo como un método eficaz, sino también como una arquitectura modular para variantes ricas del VRP; el interés se ha desplazado hacia el diseño comparable y el reporte transparente de sus componentes, al combinar: (i) destrucción parcial, (ii) reconstrucción guiada y (iii) ajuste adaptativo del uso de operadores (destroy/repair). Este marco se desarrolló originalmente para abordar problemáticas con ventanas de tiempo y, desde entonces, ha motivado múltiples extensiones y análisis sistemáticos centrados en la efectividad de los operadores y sus reglas de adaptación. (Mara *et al*., 2022; Voigt, 2025).

4.3.1. Componentes mínimos del ALNS

- *Conjunto de operadores* destroy**:** random removal, worst removal, related removal (por cercanía/tiempo), shaw removal.
- *Conjunto de operadores* repair**:** greedy insertion, regret-k insertion, time-window insertion, noise insertion.
- *Estrategia de aceptación:* SA-like (temperatura) o aceptación determinista con tolerancia; se recomienda SA-like por robustez.
- *Actualización adaptativa de pesos (operator scoring***):** recompensar operadores que mejoran o diversifican, penalizar los que estancan.
- *Mecanismos de intensificación/diversificación:* reinicios controlados, destrucciones más agresivas o perturbaciones periódicas.

Para que ALNS sea comparable entre estudios, estos componentes deben declararse como un 'contrato de ejecución': (i) conjunto exacto de operadores (destrucción y reconstrucción), (ii) regla de selección (distribución inicial y cómo cambian los pesos), (iii) criterio de aceptación (determinista o tipo recocido simulado) y calendario de temperatura si aplica, (iv) criterio de parada (tiempo o iteraciones) y (v) número de corridas/semillas. En VRP con ventanas, pequeñas decisiones —por ejemplo, cómo se computa la espera o cómo se penaliza la tardanza— pueden explicar diferencias mayores que el operador mismo, por lo que se recomienda reportarlas explícitamente (Accorsi *et al*., 2022; Mara *et al*., 2022; Voigt, 2025).

4.3.2. Diseño de operadores para ventanas y holguras

En VRP con ventanas (incluyendo ventanas blandas), los operadores deben ser "conscientes del tiempo": las inserciones deben evaluar el impacto sobre la factibilidad temporal y sobre las penalizaciones. Dos recomendaciones prácticas:

- *Evaluación incremental:* mantener los tiempos de llegada/salida y las holguras para recalcular rápidamente el costo de insertar o mover a un cliente.
- *Penalizaciones dinámicas:* ajustar la penalización por tardanza o sobrecapacidad durante la búsqueda para guiar la convergencia hacia la factibilidad.

Este tipo de decisiones de implementación deben reportarse porque afectan la comparabilidad; la literatura reciente enfatiza la transparencia de los componentes y parámetros en pruebas computacionales (Accorsi *et al*., 2022).

4.3.3. Plantilla operativa (pseudo-código) de ALNS

```
ALNS(instancia, s0, operadores D/R, parametros):
  s = s0 ; s_best = s0
  inicializar pesos w_D, w_R (uniformes)
  para iter = 1..IterMax (o hasta tiempo límite):
    elegir operador d ~ w_D ; elegir operador r ~ w_R
    s' = destroy(s, d) ; s'' = repair(s', r)
    si aceptar(s'' | s, T):
      s = s''
    actualizar s_best
    actualizar puntuaciones de d y r según calidad (mejora / aceptación /
diversidad)
    actualizar pesos (con inercia) y temperatura T (si aplica)
  retornar s_best
```

4.4. Híbridos y metaheurísticas ligeras (cuando conviene mezclar MIP con heurística)

Las metaheurísticas combinan la capacidad expresiva de los modelos MIP con la flexibilidad de las heurísticas. No se trata de resolver el VRP completo con MIP (lo cual puede resultar prohibitivo), sino de usar MIP de forma localizada: reparar partes, optimizar subrutas, seleccionar movimientos o estimar límites. Este enfoque suele ser rentable cuando existen restricciones complejas, se busca mejorar la calidad final sin rediseñar todo el motor o se requiere certidumbre parcial (por ejemplo, mejorar una ruta crítica) (Bai *et al*., 2023).

En términos operativos, estos enfoques funcionan como 'módulos de precisión': se invoca un modelo MIP en porciones del problema donde aporta mayor valor marginal (rutas críticas, ventanas estrechas, o segmentos con alta penalización), mientras que el motor metaheurístico mantiene la exploración global. Esta lógica permite elevar la calidad final sin convertir el proceso en un solucionador exacto a escala completa, lo cual es especialmente útil en VRP ricos donde el cuello de botella no es formular, sino sostener la factibilidad y el servicio bajo presupuestos de cómputo realistas (Bai *et al*., 2023).

Cuando se active un módulo MIP, debe reportarse como parte del método: tamaño del subproblema (clientes/arcos), frecuencia de activación (cada cuántas iteraciones o bajo qué disparadores), límites de tiempo del solver, criterio de aceptación del resultado del MIP y el efecto neto en costo/servicio/tiempo. Esto evita que el híbrido sea una 'caja negra' y mantiene la comparabilidad con motores puramente metaheurísticos (Accorsi *et al.,* 2022; Bai *et al*., 2023).

4.4.1. Patrones útiles de matheurística ligera

- *Fix-and-optimize:* fijar la mayor parte de la solución y optimizar con MIP una ventana de clientes/arcos (subproblema).
- Ruin-and-optimize: destruir (ruin) y luego reconstruir (optimize) con un MIP pequeño en vez de un repair heurístico.
- *Column generation local:* generar rutas candidatas mediante heurística y resolver un set partitioning reducido.
- *Warm-start:* usar la solución de ALNS/GA como arranque para MIP (si se dispone de solver).

4.4.2. Criterios prácticos para decidir "cuándo conviene"

Se sugiere activar los componentes MIP solo cuando aporten un valor marginal claro. Criterios de activación:

- Presupuesto computacional suficiente o necesidad de explicabilidad y/o garantías parciales.
- Instancias medianas con restricciones fuertes (ventanas estrechas, sincronizaciones, o alta penalización por tardanza).
- Fase final de búsqueda (post-optimization), cuando el grueso de la estructura ya es razonable.

4.5. "Learning-augmented optimization": cómo IA mejora selección de operadores / inserciones / policies

La optimización aumentada por aprendizaje propone usar modelos de ML para mejorar decisiones internas de heurísticas/metaheurísticas: seleccionar operadores, predecir costos de inserción, priorizar clientes, o aprender políticas de construcción. La idea no es reemplazar por completo el solucionador, sino inyectar información aprendida que acelere y estabilice la búsqueda (Bengio, Lodi, & Prouvost, 2021; Bai *et al*., 2023; Bogyrbayeva *et al*., 2024).

4.5.1. Niveles de integración ML → Solver

- *Predicción/estimación (plug-in):* ML estima el costo incremental, la probabilidad de tardanza o el riesgo y el solver decide.
- *Selección de operadores:* un clasificador o bandit selecciona destroy/repair según estado de la solución y señales de estancamiento.
- *Aprendizaje de políticas (end-to-end):* RL o modelos tipo GNN sugieren secuencias o inserciones; el solver refina y verifica factibilidad.
- *Aprendizaje de parámetros:* ML recomienda ajustar hiperparámetros (tuning) o realizar ajustes dinámicos (por ejemplo, penalizaciones).

4.5.2. Tendencias: GNN/RL y aprendizaje de heurísticas

Las tendencias reportadas en revisiones recientes incluyen: (i) representaciones con redes sobre grafos (GNN) para capturar estructura espacial, (ii) aprendizaje por refuerzo (RL) para construir tours o políticas de inserción, y (iii) modelos de atención para construir soluciones en forma autoregresiva. En todos los casos, se recomienda mantener un "cinturón de seguridad" heurístico: verificación de factibilidad, reparación y ajuste dinámico de rutas local (Bengio *et al*., 2021; Bai *et al*., 2023; Bogyrbayeva *et al*., 2024).

4.5.3. Evaluación justa en ML-VRP (prácticas mínimas)

Siguiendo guías de testing computacional, cualquier componente de aprendizaje automático debe evaluarse con un protocolo consistente: separación de instancias de entrenamiento y de prueba, reporte de generalización (incluyendo escenarios fuera de la distribución) y comparación con métodos de referencia sólidos bajo el mismo presupuesto computacional (Accorsi *et al*., 2022).

- Reportar datasets, seeds y splits (train/val/test).

- Comparar con ALNS/SA/TS bien afinados bajo el mismo tiempo límite.
- Reportar la huella de cómputo (GPU/CPU), los tiempos de inferencia y el costo de entrenamiento.
- Medir la robustez bajo incertidumbre (media y dispersión) cuando corresponda.

En síntesis, el aporte del aprendizaje no debe evaluarse solo por el mejor promedio, sino por (i) la estabilidad en escenarios (dispersión y percentiles), (ii) el costo total de cómputo (entrenamiento + inferencia) y (iii) la capacidad de generalizar a instancias no vistas y a condiciones cambiantes. Por ello, la comparación debe hacerse con métodos de referencia sólidos, bien afinados y ejecutados con el mismo presupuesto (Accorsi *et al*., 2022; Bengio *et al*., 2021; Bogyrbayeva *et al.,* 2024).

4.6. Arquitectura operacional: ajuste dinámico de rutas en tiempo real + integración con un Digital Twin

El salto de un solucionador offline a un sistema operacional requiere una arquitectura que soporte: el ajuste dinámico de rutas ante eventos, la observabilidad del estado, la trazabilidad de datos y las métricas de servicio. La noción de Digital Twin en logística urbana propone un marco para conectar datos, simulación y toma de decisiones, permitiendo evaluar políticas y reaccionar ante condiciones cambiantes (Belfadel *et al*., 2023; Adamo *et al*., 2024).

Desde la ingeniería del sistema, pasar de un solucionador 'offline' a operación requiere separar responsabilidades: (i) capa de datos (órdenes, ventanas, flota, telemetría), (ii) capa de estimación (ETA, congestión por franja, riesgo de no-show), (iii) capa decisional (motor de ruteo con presupuesto configurable) y (iv) capa de gobierno (reglas de negocio, SLAs, auditoría). En un gemelo digital, estas capas se acoplan mediante registros de decisiones (logs) y mecanismos de repetición del escenario, lo que permite explicar qué cambió, por qué cambió y qué impacto tuvo en el costo y el servicio (Belfadel et al., 2023; Adamo et al., 2024).

4.6.1. Ciclo operativo de ajuste dinámico de rutas

Un esquema típico incluye tres capas: planificación (plan), ejecución (do) y monitoreo/ajuste dinámico de rutas (check-act). En entornos dinámicos, la decisión no es única; se reitera ante eventos (nuevas órdenes, cancelaciones, congestión, fallas). La literatura sobre VRP dinámico-estocástico enfatiza que la política de respuesta y replanificación ante eventos debe formar parte explícita del modelo (Ulmer, 2020; Soeffker *et al.,* 2022).

- *Disparadores (triggers):* desviación de ETA, nuevas demandas, ruptura de capacidad, cliente ausente, congestión.
- *Horizonte rodante (rolling horizon):* reoptimizar cada Δt o por eventos, con congelamiento parcial de decisiones cercanas en el tiempo.
- Reparación rápida: inserciones/ALNS corto; y ajuste dinámico de rutas profunda**:** ALNS largo o metaheurística, según presupuesto.

4.6.2. Componentes mínimos de un Digital Twin para ruteo

A nivel práctico, un Digital Twin para distribución puede definirse como el acoplamiento entre: (i) un modelo de estado (vehículos, órdenes, tiempos), (ii) un módulo de simulación/estimación (tiempos de viaje, demanda, presencia), y (iii) un motor de decisión (solver) con logging completo. El valor está en la trazabilidad: qué datos llevaron a qué decisión, con qué parámetros, y con qué desempeño (Belfadel *et al*., 2023).

- *Ingesta de datos:* GPS/telemetría, órdenes, ventanas, restricciones y reglas de servicio.
- *Estimación:* modelos de tiempo de viaje dependientes del horario, predicción de no-show o de retrasos.

- *Decisión:* motor ALNS/híbrido con políticas; opción de enfoque asistido por aprendizaje automático para la selección de operadores.
- *Observabilidad:* Cuadro de mandos de SLA (tardanza, cumplimiento) y de costo; auditoría de decisiones.
- *Reproducibilidad:* versionado de instancias 'operativas' (logs) para la repetición y la evaluación comparativa interna.

Capítulo 5. Resultados, implicaciones prácticas y agenda de investigación

En este capítulo, los resultados no se presentan como una cifra aislada de "mejor costo", sino como evidencia operativa completa: calidad de la solución, nivel de servicio, tiempo computacional y confiabilidad en escenarios. Esta estructura busca impedir lecturas sesgadas y facilitar una interpretación defendible tanto en publicaciones científicas como en la adopción organizacional. Por ello, el reporte se organiza en bloques de indicadores con definiciones explícitas, unidades, reglas de cálculo y criterios de comparación. En particular, se enfatiza que, en ruteo bajo incertidumbre y con prioridades de servicio, el valor de un método se mide por su desempeño promedio y su estabilidad, así como por su capacidad para sostener acuerdos de servicio bajo condiciones variables.

5.1. Estructura de reporte de resultados (calidad, tiempo, robustez, servicio a prioritarios)

Un reporte reproducible debe separar claramente: (a) qué se optimizó y con qué presupuesto, (b) qué se obtuvo (output) y (c) qué tan estable es el desempeño cuando cambian las condiciones. Para evitar lecturas sesgadas, se recomienda reportar al menos cuatro bloques de indicadores, con definiciones explícitas, unidades, y reglas de cálculo (Accorsi *et al*., 2022).

En consecuencia, cada bloque cumple un propósito distinto: el Bloque A muestra eficiencia y estructura de la solución; el Bloque B demuestra si la eficiencia se logra sin degradar el servicio (especialmente el prioritario); el Bloque C garantiza que las comparaciones respetan el presupuesto computacional; y el Bloque D confirma si el desempeño se sostiene cuando la operación real se aparta del caso determinista.

5.1.1. Bloque A: Calidad y estructura de la solución

- Costo total desagregado: distancia/costo de viaje, penalizaciones por tardanza/holguras, costos fijos por vehículo (si aplica).
- Número de vehículos usados y longitud promedio de ruta (o carga/tiempo por vehículo).
- Indicadores de factibilidad: porcentaje de corridas factibles; violación media/máxima (capacidad, ventanas).
- Gap a la mejor referencia: relativo a mejor conocido (BKS) o al mejor método del estudio; reportar fórmula del gap.

Estos indicadores permiten distinguir tres situaciones que suelen confundirse: (i) métodos que reducen el costo, pero lo hacen a costa de una mayor fragilidad (más violaciones o mayor dispersión), (ii) métodos que logran factibilidad "a cualquier precio" inflando penalizaciones o aumentando el número de vehículos, y (iii) métodos que mejoran la eficiencia manteniendo la consistencia operacional. Por ello, el costo desagregado y la tasa de factibilidad deben leerse conjuntamente, y el "gap" debe definirse de manera transparente para evitar comparaciones ambiguas entre instancias o configuraciones. Con este propósito, la Tabla 1 resume el desempeño por instancia de cada método, reportando el promedio y el mejor valor observado (AVG [MIN]) bajo un presupuesto computacional controlado.

Tabla 1. *Desempeño por instancia: promedio y mejor valor (AVG [MIN]) por método.*

Inst	GEN AVG(MIN)	TBU AVG(MIN)	ENJ AVG(MIN)	RCC AVG(MIN)	EBL AVG(MIN)
1	175.357 (165.515)	180.653 (170.462)	207.336 (195.592)	167.407 (159.857)	213.377 (207.483)
2	274.960 (251.273)	293.873 (284.067)	326.055 (311.519)	250.671 (233.869)	333.970 (321.739)
3	155.334 (146.402)	178.857 (168.080)	180.232 (167.584)	144.757 (129.288)	182.843 (171.738)
4	134.113 (122.063)	148.851 (143.888)	157.338 (151.058)	125.256 (115.847)	159.663 (154.327)
5	133.921 (120.040)	143.299 (135.830)	160.500 (151.466)	127.809 (119.769)	167.972 (160.566)
6	186.124 (175.691)	199.952 (188.667)	209.163 (201.790)	176.751 (160.920)	214.924 (205.266)
7	105.769 (80.849)	152.007 (122.063)	261.335 (152.685)	94.407 (69.864)	302.920 (240.523)
8	47.381 (43.663)	62.240 (54.515)	68.622 (62.295)	45.323 (37.476)	72.761 (64.324)
9	65.272 (46.771)	59.604 (46.601)	86.932 (80.271)	59.376 (41.911)	99.151 (82.034)
10	70.793 (61.277)	60.890 (56.607)	86.533 (77.558)	61.083 (48.782)	90.192 (79.571)
11	49.259 (46.135)	55.001 (49.896)	59.654 (55.682)	45.368 (41.903)	61.304 (57.350)
12	162.557 (143.224)	187.595 (164.286)	217.184 (202.632)	143.715 (121.756)	222.842 (196.207)

Nota: AVG = promedio; MIN = mejor valor observado. GEN = Genético; TBU = Tabú; ENJ = Enjambre (PSO); RCC = Recocido simulado; EBL = Ébola.

5.1.2. Bloque B: Nivel de servicio y priorización

En aplicaciones reales la prioridad no es homogénea: existen clientes críticos, entregas con promesa horaria estricta o segmentos de alto impacto. Por tanto, además del costo se deben reportar SLA y métricas por segmentos (prioritarios vs no prioritarios).

- Cumplimiento de ventana: % on-time, tardanza media (min), tardanza P90/P95 y tardanza máxima.
- Métrica de prioridad: tasa de cumplimiento ponderada por prioridad (p_i), o tardanza ponderada.
- Reintentos/no-show (si aplica): % visitas fallidas y costo de reentrega (Özarık et al., 2021).
- Equidad de servicio: dispersión de tardanza entre zonas/grupos (opcional, pero recomendado actualmente).

Dado que el modelo incorpora prioridades de servicio, reportar un único SLA agregado puede ocultar degradaciones relevantes. Por tanto, estas métricas se calculan por segmentos (prioritarios y no prioritarios) y, cuando corresponda, también por zonas o grupos operativos. Esta segmentación permite identificar si una mejora de costos se mantiene sin deteriorar el cumplimiento con clientes críticos, donde usualmente se concentra el riesgo reputacional y contractual.

5.1.3. Bloque C: Tiempo computacional y presupuesto de cómputo

Para comparaciones válidas, el tiempo debe reportarse como desempeño bajo un presupuesto controlado, con detalles del entorno (hardware, hilos, versiones de las librerías). Se recomienda reportar tanto el tiempo de pared (wall-clock) como el número de evaluaciones u operaciones clave, especialmente en métodos estocásticos (Accorsi *et al*., 2022).

- Tiempo medio/mediana por instancia y por corrida; tiempo total de experimentación.
- Presupuesto de búsqueda: iteraciones, evaluaciones de vecindario, o número de llamadas a repair/destroy (ALNS).
- Entorno: CPU/GPU, RAM, número de hilos, sistema operativo, versión del solver (si aplica).
- Costo de entrenamiento e inferencia (en métodos con ML), incluyendo los tiempos de inferencia por decisión.

Tabla 2. *Tiempo de ejecución por instancia — Promedio (Mejor valor) (AVG = promedio; MIN = mejor valor observado; en las mismas unidades reportadas en el proyecto experimental.)*

Instancia	Genético (AVG [MIN])	Tabú (AVG [MIN])	Enjambre/PSO (AVG [MIN])	Recocido (AVG [MIN])	Ébola (AVG [MIN])
1	3.402 (2.859)	901 (713)	1.520 (1.317)	1.267 (1.029)	1.689 (1.463)
2	2.444 (2.056)	734 (578)	1.136 (973)	957 (763)	1.265 (1.080)
3	2.156 (1.782)	742 (585)	1.091 (905)	944 (747)	1.247 (1.035)
4	2.007 (1.707)	758 (607)	1.046 (895)	914 (727)	1.198 (1.009)
5	1.995 (1.698)	741 (598)	1.065 (919)	921 (731)	1.203 (1.011)
6	2.336 (1.928)	780 (617)	1.141 (985)	981 (792)	1.321 (1.121)
7	2.110 (1.730)	800 (646)	1.098 (935)	940 (765)	1.244 (1.080)
8	1.712 (1.386)	662 (530)	961 (825)	821 (663)	1.068 (886)
9	1.751 (1.397)	670 (528)	992 (860)	819 (658)	1.097 (893)
10	1.799 (1.454)	685 (547)	1.009 (875)	845 (686)	1.116 (927)
11	1.698 (1.365)	650 (520)	951 (829)	801 (650)	1.050 (870)
12	2.287 (1.910)	770 (616)	1.122 (967)	971 (781)	1.309 (1.110)

Para garantizar comparaciones válidas, el tiempo se reporta como tiempo de ejecución real (tiempo de pared) bajo un presupuesto controlado. En métodos estocásticos, además del tiempo total, se recomienda reportar el volumen de búsqueda (iteraciones, evaluaciones u operaciones clave), ya que el tiempo por sí solo puede variar por implementación.

5.1.4. Bloque D: Robustez y confiabilidad bajo escenarios

Cuando existe incertidumbre (demanda, servicio, viaje, presencia), el resultado debe medirse como una distribución y no como un valor único. Se recomiendan métricas de media y dispersión, y al menos un indicador de riesgo (percentiles o CVaR) (Ulmer, 2020; Soeffker *et al*., 2022).

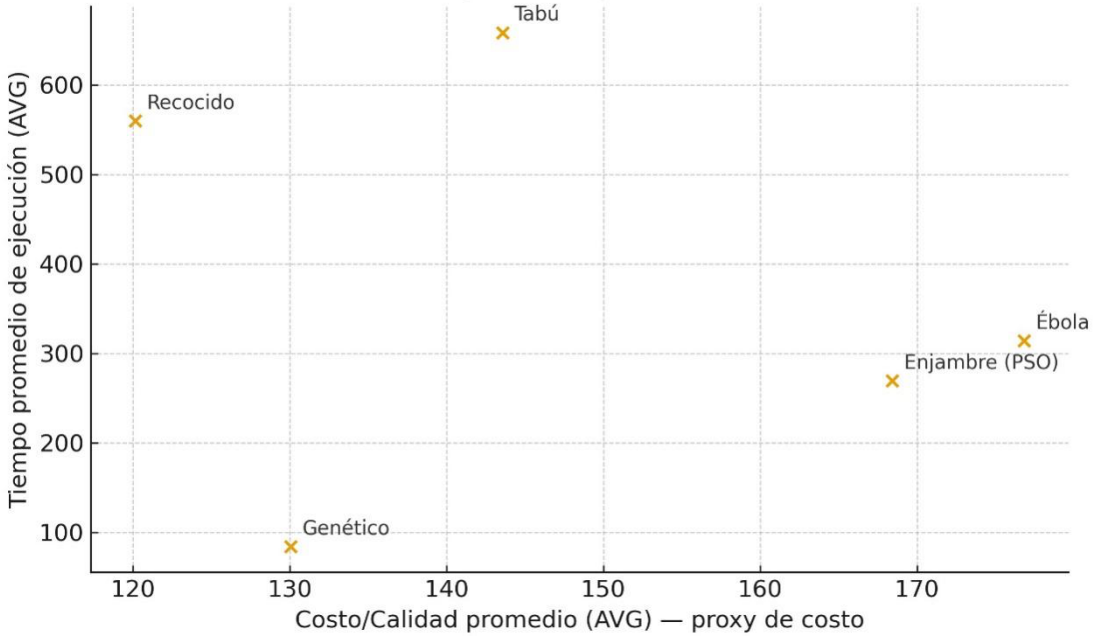
- E[C], SD(C) y coeficiente de variación; percentiles (P50, P90, P95).
- Tasa de factibilidad bajo escenarios (%), y tasa de cumplimiento de SLA bajo escenarios (P90 SLA).
- Sensibilidad: curvas costo–servicio al variar penalizaciones, flexibilidad de ventanas, o congestión.
- Estabilidad operacional: número de cambios de ruta respecto al plan base (si se reoptimiza).

En presencia de incertidumbre, el desempeño debe interpretarse como una distribución. Por eso, además de promedios, se reportan dispersiones y percentiles, y se visualiza la variabilidad mediante gráficos que permitan comparar estabilidad entre métodos. Un método se considera más confiable cuando mantiene un buen desempeño no solo “en promedio”, sino también en escenarios exigentes (colas de la distribución).

Para complementar la lectura sobre distribución y dispersión, resulta útil observar la compensación entre el desempeño y el esfuerzo computacional. En particular, una solución puede mejorar el costo-calidad promedio, pero a costa de un mayor tiempo de ejecución, lo que condiciona su viabilidad en operación cuando existen presupuestos de cómputo y la necesidad

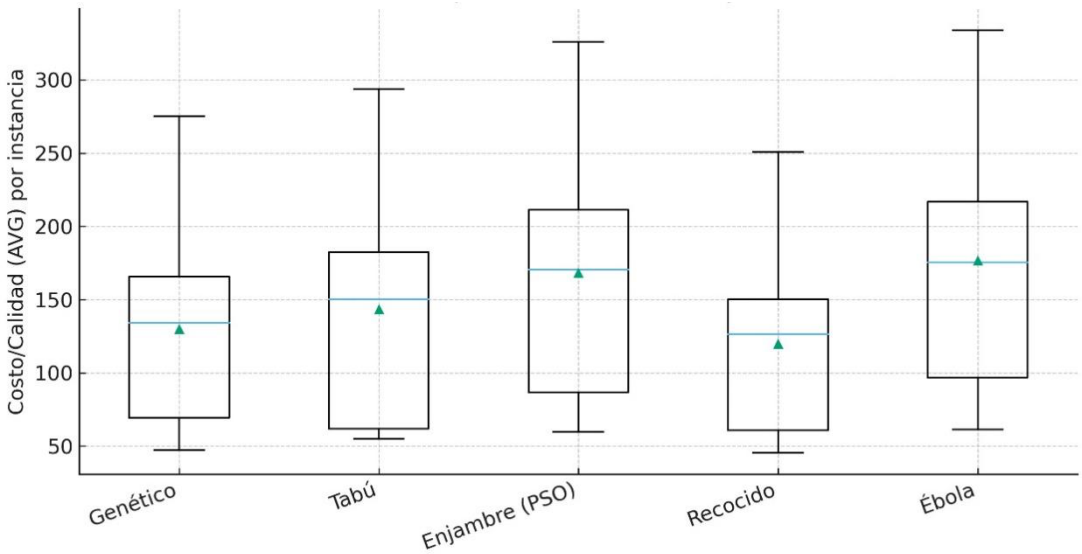
de respuesta rápida. Por ello, la Figura 1 presenta una frontera costo–tiempo (como proxy de la relación costo–servicio cuando no se dispone de métricas SLA por método), lo que permite identificar puntos de equilibrio en los que las mejoras marginales dejan de justificar incrementos significativos en el tiempo.

Figura 1. Frontera costo–tiempo (proxy de compensación costo–servicio)



Dado que el modelo opera bajo incertidumbre y emplea métodos estocásticos, el desempeño debe interpretarse como una distribución. Una forma directa de visualizar la estabilidad relativa entre métodos es comparar la dispersión del costo-calidad a lo largo de las instancias, como se muestra en la Figura 2.

Figura 2. Distribución del costo/calidad (AVG) por instancia.



Nota: el boxplot resume la variabilidad del desempeño entre instancias; para "robustez bajo escenarios" (semillas), se requiere el registro por corrida.

5.2. Lectura gerencial: compensaciones costo–servicio–confiabilidad (qué decisiones cambian el resultado)

En el contexto organizacional, los resultados deben traducirse a decisiones: cuánto penalizar la tardanza, qué tolerancia aceptar en ventanas flexibles, cuánto priorizar clientes críticos y cada cuánto ajustar el plan ante eventos. Por ello, el apartado interpreta los resultados como equilibrios entre costo, servicio y confiabilidad, identificando qué decisiones mueven significativamente dichos equilibrios y cuáles producen mejoras marginales. Esta lectura evita tratar la optimización como un fin abstracto y la convierte en un insumo para definir políticas operativas y de servicio coherentes con la realidad del negocio.

Desde la gerencia, la pregunta no es solo "qué algoritmo ganó", sino también qué decisiones de diseño y operación explican la diferencia. Esto requiere traducir parámetros técnicos (penalizaciones, frecuencia de ajuste dinámico de rutas, elección de operadores) a decisiones de negocio: promesas de servicio, capacidad instalada, reglas de asignación y tolerancia al riesgo (Bai *et al*., 2023; Adamo *et al*., 2024).

5.2.1. Compensación 1: costo vs nivel de servicio

Al aumentar la penalización por tardanza o estrechar ventanas, el sistema tiende a usar más vehículos, recorrer más distancia o esperar más, para sostener SLA. Este intercambio debe representarse como una frontera costo–servicio y acompañarse de escenarios de decisión (p. ej., 'SLA del 95% con +3% de costo').

- Decisión clave: ponderación de penalizaciones (π) frente a costo de viaje.
- Efecto típico: mayor SLA → más vehículos / más distancia / mayor espera planificada.
- Recomendación: fijar el SLA objetivo y optimizar el costo sujeto a ese SLA (o usar un multiobjetivo).

5.2.2. Compensación 2: eficiencia vs robustez (variabilidad)

Soluciones muy ajustadas (casi sin holgura) pueden ser baratas en promedio, pero frágiles ante la congestión o las demoras. Aumentar holgura u optimizar riesgo (p. ej., CVaR) suele elevar costo medio, pero reduce colas de tardanza y fallas de SLA (Ulmer, 2020; Soeffker *et al*., 2022).

- Decisión clave: objetivo robusto (λ·Var o λ·CVaR) y tamaño del conjunto de escenarios.
- Efecto típico: menor SD y P95 de tardanza, con ligero aumento del costo medio.
- Recomendación: reportar P90/P95 de SLA; no basarse solo en promedios.

5.2.3. Compensación 3: planificación offline vs ajuste dinámico de rutas online

Reoptimizar con horizonte rodante reduce tardanzas y fallas, pero introduce "inestabilidad" (cambios frecuentes de plan). Por eso, se recomienda medir la estabilidad (número de reasignaciones y cambios) y definir una política de congelamiento (freeze) (Ulmer, 2020).

- Decisión clave: frecuencia Δt de ajuste dinámico de rutas y regla de freeze.
- Efecto típico: más reoptimizaciones → mejor SLA pero más cambios y complejidad operativa.
- Recomendación: ajuste dinámico de rutas por eventos + umbrales (triggers) para evitar ruido.

5.2.4. Qué decisiones 'mueven la aguja' (resumen ejecutivo)

- Definición de prioridades y SLAs por segmento (qué se protege primero).
- Penalizaciones y tolerancias (soft windows vs hard windows).
- Presupuesto de cómputo (30s, 2min, 10min) y momento de usar intensificación.
- Selección y peso de operadores (en ALNS) y estrategia de aceptación (temperatura).
- Datos de viaje (estáticos vs dependientes del tiempo) y política de ajuste dinámico de rutas.

5.3. Guía de implementación en empresa (datos necesarios, telemetría, reglas de negocio, SLAs)

La transferencia desde un prototipo académico a la operación requiere datos consistentes, integración técnica y gobernanza de las reglas. Se recomienda un enfoque incremental: (i) piloto offline con datos reales, (ii) operación asistida, (iii) operación semi-automática con auditoría y monitoreo, y (iv) ajuste dinámico de rutas en tiempo real. El enfoque de Digital Twin ayuda a trazar decisiones y resultados (Belfadel *et al*., 2023).

5.3.1. Datos mínimos (sin los cuales el modelo falla)

- Clientes: geocodificación confiable, dirección normalizada, zona, prioridad/segmento, ventanas base y tolerancias.
- Demanda y servicio: unidades (kg, cajas, paradas), tiempos de servicio por tipo, y reglas (firma, verificación, etc.).
- Flota: capacidad, restricciones (refrigerado, tamaño, turnos), costos (fijo/variable) y disponibilidad.
- Depósitos y hubs: horarios, capacidad de carga, colas y tiempos de preparación.
- Red de viaje: matriz de distancias y tiempos; idealmente, dependiente de la hora y actualizable (Adamo *et al*., 2024).

5.3.2. Telemetría y eventos (para operación 2025+)

- GPS y estado del vehículo (posición, inicio/fin de servicio, paradas, incidencias).
- Tiempos de viaje observados vs previstos (calibración del modelo).
- Eventos de cliente: ausencia/no-show, cambios de ventana, cancelaciones.
- Calidad de datos: gestión de outliers, retrasos sistemáticos y sesgos por zona.

5.3.3. Reglas de negocio típicas y parametrización

- Priorización: clientes críticos (p_i), penalizaciones diferenciadas por segmento.
- Explicabilidad: reglas de reasignación (qué cambios se permiten y cuándo).
- Congelamiento: no modificar las próximas X paradas, salvo emergencia.
- Políticas de reintento: límite de reentregas, ventana alternativa, costos por reintento.
- SLAs: % on-time objetivo por segmento; tolerancias por tardanza y compensaciones.

5.3.4. Integración técnica (mínimo viable)

- ETL de datos: extracción, limpieza, normalización y validación diaria.
- Motor de ruteo: servicio API (solve/offline + reopt/online) con logs completos.
- Monitoreo: tablero SLA y costo; alertas por riesgo de tardanza.
- Auditoría: versionado de instancias y decisiones (replay).
- Ciclo de mejora: calibración de tiempos y revisión mensual de reglas/SLAs.

5.4. Limitaciones y próximos pasos (multi-producto, sostenibilidad, automatización, equidad de servicio)

Como buena práctica en trabajos contemporáneos, se sugiere declarar explícitamente las amenazas a la validez (calidad de los datos, generalización, presupuesto de cómputo y sensibilidad a hiperparámetros) y delinear una agenda de investigación que articule las restricciones de operación con desarrollos metodológicos recientes (Accorsi *et al*., 2022; Bai *et al.,* 2023; Bogyrbayeva et *al.,* 2024; Voigt, 2025).

5.4.1. Limitaciones típicas

- Representación simplificada del viaje (matriz estática), aunque haya congestión y dependencia del tiempo.
- Hipótesis de distribución/escenarios: independencia, estacionariedad o tamaño limitado de la muestra.
- Sensibilidad a hiperparámetros (ALNS/GA/PSO) y riesgo de overfitting en el experimento.
- Costos omitidos: tiempos de carga/descarga, restricciones laborales finas, restricciones de seguridad.
- Generalización: instancias sintetizadas que no capturan todos los patrones de la empresa.

5.4.2. Agenda de investigación emergente (líneas priorizadas)

- Multi-producto y restricciones ricas: compatibilidad, temperatura, apilamiento y multi-depósito con hubs intermedios.
- Sostenibilidad: objetivos de emisiones y energía, y transición a EVRP con infraestructura de carga (Küçükoğlu *et al*., 2021).
- Automatización: tuning automático, selección de operadores y pipelines de evaluación estandarizados (Accorsi *et al*., 2022; Voigt, 2025).
- Equidad de servicio: métricas y restricciones para evitar sesgos por zona/segmento (fairness operacional).
- Learning-augmented: ML para anticipar la congestión y los no-shows y para guiar repair/destroy con garantías metodológicas (Bai *et al*., 2023; Bogyrbayeva *et al*., 2024).
- Última milla híbrida (incl. drones) cuando sea viable y regulatoriamente permitida (Jazairy *et al.,* 2025).

Glosario (términos operativos y de optimización)

Aceptación (criterio de): Regla que decide si una solución candidata reemplaza a la solución actual (p. ej., aceptación tipo SA que permite empeorar temporalmente para escapar de óptimos locales).

ALNS (Búsqueda adaptativa en vecindarios grandes): Marco metaheurístico basado en destruir parcialmente una solución y repararla, ajustando adaptativamente la probabilidad de uso de operadores según su desempeño histórico.

Arco (i,j): Conexión dirigida entre dos nodos del grafo; representa un viaje posible desde i hacia j con un costo/tiempo asociado.

Baseline (línea base): Método de referencia mínimo y reproducible usado para comparar algoritmos; debe tener presupuesto y reglas de evaluación claramente definidos.

Benchmarking: Proceso sistemático de comparación entre métodos bajo condiciones controladas (instancias, presupuesto, semillas, hardware, métricas y reporte).

Capacidad (Q): Límite de carga del vehículo (en unidades, kg, volumen u otro recurso); genera restricciones de factibilidad en rutas.

Cliente prioritario: Cliente con mayor peso de servicio (p_i) o mayor penalización por tardanza; determina métricas y objetivos orientados a SLA.

Costo incremental (Δc): Cambio de costo al insertar o mover un cliente en una ruta; se usa en heurísticas de inserción y reparaciones.

Cobertura: Medida del conjunto de clientes atendidos o atendidos en un umbral de servicio; se usa cuando hay restricciones de servicio parcial o presencia incierta.

Congelamiento (freeze): Política operacional que evita modificar decisiones cercanas en el tiempo (p. ej., próximas k paradas), aun si se reoptimiza, para preservar estabilidad.

CVaR (riesgo): Métrica de riesgo que captura el promedio de los peores casos (cola) de una distribución de costos o tardanzas; útil en robustez.

Demanda (d_i): Requerimiento del cliente i; puede ser determinista o aleatorio (por escenarios o distribución).

Destroy/Repair: Operadores de ALNS. *Destroy* elimina parte de la solución (clientes o arcos) y *Repair* reconstruye insertándolos de nuevo de forma guiada.
Desviación estándar (SD): Medida de dispersión; en robustez, cuantifica variabilidad de costo o tardanza bajo múltiples corridas/escenarios.
Evento operacional: Suceso que obliga a reoptimizar o ajustar el plan (nueva orden, cancelación, congestión, vehículo inactivo, cliente ausente).
Factibilidad: Propiedad de una solución que cumple todas las restricciones (capacidad, ventanas, etc.). En enfoques con penalización puede permitirse infracción temporal controlada.
Frontera costo–servicio (trade-off): Relación entre costo total y nivel de servicio (SLA). Permite identificar puntos de equilibrio y decisiones gerenciales.
Giant tour: Representación donde los clientes aparecen en una permutación global y luego se "segmenta" en rutas factibles mediante un procedimiento *split*.
Holgura (slack): Tiempo disponible antes de violar un límite (p. ej., b_i − $llegada_i$). Mayor holgura suele implicar mayor robustez.
Horizonte rodante (rolling horizon): Estrategia de ajuste dinámico de rutas periódica o por eventos; replantea decisiones conforme avanza el tiempo.
Inferencia (ML): Tiempo de ejecución del modelo de aprendizaje para producir una recomendación (p. ej., elegir operador, estimar costo o priorizar inserción).
Instancia: Conjunto de datos que define un problema a resolver (clientes, depósitos, demandas, distancias, ventanas, flota, etc.).
Inserción (greedy / regret-k): Heurísticas de reparación: *greedy* elige la inserción de menor costo incremental; *regret-k* prioriza clientes con alto "arrepentimiento" si no se insertan pronto.
No-show: Ausencia del cliente al momento del servicio; induce reintentos, penalizaciones o replanificación y afecta SLA/robustez.
Objetivo robusto: Función objetivo que penaliza variabilidad o riesgo, no solo promedio (p. ej., $E[C] + \lambda \cdot SD(C)$ o CVaR).
Penalización: Término que convierte una violación (tardanza, sobrecapacidad) en costo para guiar la búsqueda hacia factibilidad o hacia SLAs.
Presupuesto computacional: Límite de tiempo o número de iteraciones/evaluaciones permitido; debe ser idéntico al comparar métodos.
Ajuste dinámico de rutas: Ajuste del plan de rutas durante la ejecución, como respuesta a eventos o de forma periódica; puede incluir reglas de freeze.
Riesgo operacional: Probabilidad de incumplir SLA o inviabilizar rutas por incertidumbre (tráfico, demanda, servicio), medible con percentiles o CVaR.
Ruta: Secuencia ordenada de visitas asignadas a un vehículo, iniciando (y usualmente terminando) en el depósito.
SLA (nivel de servicio): Compromiso medible; típicamente % de entregas on-time, tardanza P95, y reglas diferenciadas por segmento.
Soft time windows (ventanas blandas): Ventanas de tiempo que pueden violarse pagando una penalización (tardanza o earliness), a diferencia de ventanas duras.
Split: Procedimiento que particiona una permutación (giant tour) en rutas cumpliendo restricciones, minimizando costo o penalización.
Tiempo dependiente del tiempo (time-dependent): Tiempo de viaje que cambia según la hora del día; relevante en contextos urbanos con congestión.
Tuning de parámetros: Ajuste de hiperparámetros (temperatura, pesos, tasas de operador) por búsqueda sistemática o automatizada; debe separarse de la prueba final.
Ventana de tiempo [a_i, b_i]: Intervalo en el cual puede (o debe) iniciarse el servicio al cliente i; puede incluir tolerancias o múltiples ventanas.

Lista de siglas

ALNS: *Adaptive Large Neighborhood Search* (Búsqueda adaptativa en vecindarios grandes).
API: *Application Programming Interface* (Interfaz de programación de aplicaciones).
BKS: *Best Known Solution* (Mejor solución conocida).
CVaR: *Conditional Value at Risk* (Valor en Riesgo Condicional).
EDF: *Earliest Due Date* (Fecha límite más temprana).
EVRP: *Electric Vehicle Routing Problem* (Problema de ruteo de vehículos eléctricos).
GA: *Genetic Algorithm* (Algoritmo genético).
GNN: *Graph Neural Network* (Red neuronal sobre grafos).
GPU: *Graphics Processing Unit* (Unidad de procesamiento gráfico).
KPI: *Key Performance Indicator* (Indicador clave de desempeño).
LNS: *Large Neighborhood Search* (Búsqueda en vecindarios grandes).
MIP/MILP: *Mixed-Integer (Linear) Programming* (Programación entera mixta / lineal).
ML: *Machine Learning* (Aprendizaje automático).
PSO: *Particle Swarm Optimization* (Optimización por enjambre de partículas; cada partícula representa una solución candidata).
RL: *Reinforcement Learning* (Aprendizaje por refuerzo).
SA: *Simulated Annealing* (Recocido simulado).
SLA: *Service Level Agreement* (Acuerdo de nivel de servicio).
SD: *Standard Deviation* (Desviación estándar).
TDVRP: *Time-Dependent Vehicle Routing Problem* (VRP dependiente del tiempo).
TS: *Tabu Search* (Búsqueda tabú).
VRP: *Vehicle Routing Problem* (Problema de ruteo de vehículos).
VRPTW: *Vehicle Routing Problem with Time Windows* (VRP con ventanas de tiempo).
SFMVRPMSTWTDCP: **Enfoque propuesto de modelación y solución**: *Stochastic Flexible Multi-Vehicle Routing Problem with Multi Soft Time Windows, Time-Dependent Travel, and Delivery/Coverage Prioritization*; es decir, un VRP multi-vehículo con incertidumbre, flexibilidad operativa, ventanas de tiempo blandas múltiples, tiempos de viaje dependientes del tiempo y priorización de entregas/cobertura según criterios de servicio.

Referencias

Accorsi, L., Lodi, A., & Vigo, D. (2022). Guidelines for the computational testing of machine learning approaches to vehicle routing problems. *Operations Research Letters, 50*(2), 229–234. https://doi.org/10.1016/j.orl.2022.01.018

Adamo, T., Gendreau, M., Ghiani, G., & Guerriero, E. (2024). A review of recent advances in time-dependent vehicle routing. *European Journal of Operational Research, 319*(1), 1–15. https://doi.org/10.1016/j.ejor.2024.06.016

Bai, R., Chen, X., Chen, Z.-L., Cui, T., Gong, S., He, W., Jiang, X., Jin, H., Jin, J., Kendall, G., Li, J., Lu, Z., Ren, J., Weng, P., Xue, N., & Zhang, H. (2023). Analytics and machine learning in vehicle routing research. *International Journal of Production Research, 61*(1), 4–30. https://doi.org/10.1080/00207543.2021.2013566

Belfadel, A., Hörl, S., Tapia, R. J., Politaki, D., Kureshi, I., Tavasszy, L., & Puchinger, J. (2023). A conceptual digital twin framework for city logistics. *Computers, Environment and Urban Systems, 103*, 101989. https://doi.org/10.1016/j.compenvurbsys.2023.101989

Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research, 290*(2), 405–421. https://doi.org/10.1016/j.ejor.2020.07.063

Bjelić, N., Vidović, M., Popović, D., & Ratković, B. (2022). *Rolling-horizon approach in solving dynamic multisize multi-trailer container drayage problem*. Expert Systems with Applications, 201, 117170. https://doi.org/10.1016/j.eswa.2022.117170

Bogyrbayeva, A., Meraliyev, M., Mustakhov, T., & Dauletbayev, B. (2024). Machine learning to solve vehicle routing problems: A survey. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2023.3334976

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: A survey from an operational research perspective. *OR Spectrum, 43*, 1–58. https://doi.org/10.1007/s00291-020-00607-8

Escudero-Santana, A., Muñuzuri, J., Lorenzo-Espejo, A., & Muñoz-Díaz, M.-L. (2022). Improving e-commerce distribution through last-mile logistics with multiple possibilities of deliveries based on time and location. *Journal of Theoretical and Applied Electronic Commerce Research, 17*(2), 507–521. https://doi.org/10.3390/jtaer17020027

Gmira, M., Gendreau, M., Lodi, A., & Potvin, J.-Y. (2021). *Tabu search for the time-dependent vehicle routing problem with time windows on a road network*. European Journal of Operational Research, 288(1), 129–140. https://doi.org/10.1016/j.ejor.2020.05.041

Gunawan, A., Kendall, G., McCollum, B., Seow, H.-V., y Lee, L. S. (2021). *Vehicle routing: Review of benchmark datasets*. Journal of the Operational Research Society, 72(8), 1794–1807. https://doi.org/10.1080/01605682.2021.1884505

Gutiérrez, A., (et al.). (2024). *A Two-echelon Vehicle Routing Problem with time-dependent travel times in the city logistics context*. EURO Journal on Transportation and Logistics, 13, 100133. https://doi.org/10.1016/j.ejtl.2024.100133

Houssein, E. H., Saeed, M. K., Hu, G., & Al-Sayed, M. M. (2024). *Metaheuristics for solving global and engineering optimization problems: Review, applications, open issues and challenges*. *Archives of Computational Methods in Engineering, 31*, 4485–4519. https://doi.org/10.1007/s11831-024-10168-6

Jastrzab, T., Myller, M., Tulczyjew, L., Blocho, M., Kawulok, M., Czornik, A., y Nalepa, J. (2024). *Standardized validation of vehicle routing algorithms*. Applied Intelligence, 54, 1335–1364. https://doi.org/10.1007/s10489-023-05212-0

Jazairy, A., Persson, E., Brho, M., von Haartman, R., & Hilletofth, P. (2025). Drones in last-mile delivery: A systematic literature review from a logistics management perspective. *The International Journal of Logistics Management, 36*(7), 1–62. https://doi.org/10.1108/IJLM-04-2023-0149

Keskin, M., Branke, J., & Deineko, V. (2023). *Dynamic multi-period vehicle routing with touting*. European Journal of Operational Research, 310(3), 168–184. https://doi.org/10.1016/j.ejor.2023.02.037

Küçükoğlu, İ., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering, 161*, 107650. https://doi.org/10.1016/j.cie.2021.107650

Le, T. V., & Fan, R. (2024). *Digital twins for logistics and supply chain systems: Literature review, conceptual framework, research potential, and practical challenges*. Computers & Industrial Engineering, 187, 109768. https://doi.org/10.1016/j.cie.2023.109768

Mara, S. T. W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., & Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research, 146*, 105903. https://doi.org/10.1016/j.cor.2022.105903

Martín-Santamaría, R., Cavero, S., Herrán, A., Duarte, A., & Colmenar, J. M. (2024). *A Practical Methodology for Reproducible Experimentation: An Application to the Double-Row Facility Layout Problem*. Evolutionary Computation, 32(1), 69–104. https://doi.org/10.1162/evco_a_00317

Merchán, D., Arora, J., Pachon, J., Konduri, K., Winkenbach, M., Parks, S., y Noszek, J. (2024). *2021 Amazon Last Mile Routing Research Challenge: Data set*. Transportation Science, 58**(**1), 8–11. https://doi.org/10.1287/trsc.2022.1173

Nucamendi-Guillén, S., Gómez-Padilla, A., Olivares-Benítez, E., & Moreno-Vega, J. M. (2021). *The multi-depot open location routing problem with a heterogeneous fixed fleet*. *Expert Systems with Applications, 165*, 113846. https://doi.org/10.1016/j.eswa.2020.113846

Ojeda Rios, B. H., Xavier, E. C., Miyazawa, F. K., Amorim, P., Curcio, E., & Santos, M. J. (2021). Recent dynamic vehicle routing problems: A survey. *Computers & Industrial Engineering, 160*, 107604. https://doi.org/10.1016/j.cie.2021.107604

Özarık, S. S., Veelenturf, L. P., Van Woensel, T., & Laporte, G. (2021). Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence. *Transportation Research Part E: Logistics and Transportation Review, 148*, 102263. https://doi.org/10.1016/j.tre.2021.102263

Røpke, S., & Pisinger, D. (2006). *An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows*. Transportation Science, 40(4), 455–472. https://doi.org/10.1287/trsc.1050.0135

Soeffker, N., Ulmer, M. W., & Mattfeld, D. C. (2022). Stochastic dynamic vehicle routing in practice: Model, algorithms and case insights. *European Journal of Operational Research, 298*(3), 801–820. https://ideas.repec.org/a/eee/ejores/v298y2022i3p801-820.html

Ulmer, M. W. (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics, 9*, 100008. https://doi.org/10.1016/j.ejtl.2020.100008

Vidal, T., Laporte, G., y Matl, P. (2020). *A concise guide to existing and emerging vehicle routing problem variants*. European Journal of Operational Research, 286(2), 401–416. https://doi.org/10.1016/j.ejor.2019.10.010

Voigt, S. (2025). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research, 322*(2), 357–375. https://doi.org/10.1016/j.ejor.2024.05.033

Wu, Q., Xia, X., Song, H., Xu, X., Zhang, Y., Yu, F., y Wu, H. (2024). *A neighborhood comprehensive learning particle swarm optimization for the vehicle routing problem with time windows*. Swarm and Evolutionary Computation, 84, 101425. https://doi.org/10.1016/j.swevo.2023.101425

Xu, K., Zhang, Y., & Chen, X. (2020). Multi-vehicle routing problems with soft time windows: Models and solution insights. *Transportation Research Part C: Emerging Technologies, 120*, 102861. https://doi.org/10.1016/j.trc.2020.102861

Xu, R., Li, S., y Wu, J. (2023). *Multi-Trip Vehicle Routing Problem with Time Windows and Resource Synchronization on Heterogeneous Facilities*. Systems, 11(8), 412. https://doi.org/10.3390/systems11080412