

Índice

1. Introducción

- Presentación del problema
- Definiciones clave
- Revisión bibliográfica
- Estructura de la tesis

2. Recolección y entrega con incompatibilidades

- Definición formal del problema
- Solución propuesta
- Heurísticas (Swap, 3-opt, VNS)
- Implementación y experimentación (Instancias y entorno, Solución inicial, Detalles de implementación, Optimizaciones)

3. Conclusiones y trabajo futuro

- Conclusiones
- Trabajo futuro

4. Bibliografía y Apéndices

- Resultados
- Código

Introducción

Heurísticas para el problema de recolección y entrega punto a punto con incompatibilidades

El Problema de Recolección y Entrega Punto a Punto con Distancias Asimétricas e Incompatibilidades (PDPwI, por sus siglas en inglés) es una extensión del Problema de Recolección y Entrega Punto a Punto con Distancias Asimétricas (APDP). Consiste en planificar la ruta de un único vehículo que debe recoger y entregar objetos entre pares de ubicaciones, respetando restricciones de precedencia y compatibilidad. A diferencia del APDP, el PDPwI incorpora restricciones de incompatibilidad: ciertos objetos no pueden ser transportados conjuntamente debido a condiciones específicas, como requerimientos de temperatura, seguridad o normativas regulatorias. Esta variante incrementa significativamente la complejidad del problema, vinculándolo con otros desafíos de optimización combinatoria, como el coloreo de grafos o el problema de empaquetamiento.

El presente trabajo se enfoca en la resolución del PDPwI mediante enfoques heurísticos, con el objetivo de encontrar soluciones de buena calidad en tiempos computacionales razonables. Mientras que investigaciones previas han abordado este problema utilizando métodos exactos, como algoritmos de Branch and Cut, aquí se

propone una estrategia basada en Variable Neighborhood Search (VNS). Esta técnica se compone de dos heurísticas específicas —una variante de 3-opt y otra que denominamos “swap”—, la primera permite profundizar en la exploración del espacio de soluciones mientras que la segunda otorga variabilidad.

Para evaluar el desempeño de los métodos propuestos, se construyó un conjunto de instancias basado en las definiciones presentadas previamente por otros trabajos y adaptándolas a nuestras necesidades. Dado que las instancias originales no se encuentran disponibles públicamente, las recreamos siguiendo los criterios y procedimientos especificados en dichos trabajos.

Palabras clave: *Problema de recolección y entrega, grafo de incompatibilidades, problema de ruteo, Optimización, NP-difícil, VNS*

Presentación del problema

El Problema de Recolección y Entrega Punto a Punto (One-to-One Pickup and Delivery Problem, o PDP) consiste en planificar la ruta de un vehículo que parte desde un depósito inicial, realiza un conjunto de pedidos y finaliza su recorrido en un depósito final, que puede coincidir o no con el depósito de partida. Cada pedido está compuesto por un par de ubicaciones: un vértice de recolección (pickup) y un vértice de entrega (delivery). La ruta debe respetar la precedencia entre estos pares —es decir, cada objeto debe ser recogido antes de ser entregado—, y el objetivo es minimizar la distancia total recorrida.

Existen múltiples variantes del PDP que incorporan restricciones adicionales. Por ejemplo, en el Pickup and Delivery Problem with Time Windows (PDPTW) se exige que los pedidos se completen dentro de ciertas ventanas horarias. También existen variantes que consideran múltiples vehículos, con o sin capacidad limitada, compartimentos internos (por ejemplo, para productos refrigerados o peligrosos), o restricciones de carga y descarga. A su vez, los problemas pueden definirse sobre grafos simétricos o asimétricos, dependiendo de si la distancia desde un vértice v a otro w es igual o no a la distancia inversa.

En este trabajo se aborda el Problema de Recolección y Entrega Punto a Punto con Incompatibilidades y distancias simétricas (PDPwI), introducido por Pablo Factorovich (Factorovich, Pablo. 2020). Esta variante extiende el modelo clásico al incorporar restricciones de incompatibilidad entre objetos: ciertos ítems no pueden compartir el vehículo en simultáneo. En dicho trabajo se demuestra que el PDPwI no solo generaliza al APDP, sino también a otros problemas clásicos como el Vertex Coloring Problem, el Bin Packing Problem y el Bin Packing Problem with Conflicts. Por lo tanto, es NP-difícil.

Estas restricciones de ruteo emergen en escenarios reales donde deben transportarse productos incompatibles, como alimentos junto con productos químicos, o pasajeros con requerimientos específicos que hacen que puedan viajar juntos. Una forma de abordar este desafío consiste en utilizar múltiples vehículos o vehículos especialmente diseñados con compartimentos aislados. No obstante, estas soluciones implican mayores costos operativos o de infraestructura. El enfoque propuesto, que retomamos en este trabajo, consiste en planificar una única ruta que minimice el recorrido total y asegure que no coexistan productos incompatibles dentro del vehículo durante el trayecto.

Definiciones claves

Dentro de esta tesis utilizaremos términos que hacen referencia a algoritmos, técnicas, etc por lo que dentro de esta sección explicaremos algunos para dar una lectura más sencilla.

- **2-opt:** es una heurística de mejora local ampliamente utilizada en problemas de optimización de rutas. Su operador toma una ruta existente, elimina dos aristas y reconecta los nodos -idealmente- de manera más eficiente, reduciendo la longitud total del recorrido.
- **3-opt:** es una heurística de mejora local utilizada en problemas de optimización de rutas. Es una extensión de 2-opt, y su operador intenta reducir la distancia total de una ruta eliminando tres aristas y reconectando los segmentos de una manera más eficiente. Existen diferentes formas de reconectar los segmentos, dependiendo del problema solo algunas son factibles.
- **Algoritmo branch & cut:** es una técnica utilizada para resolver problemas de optimización combinatoria. Este algoritmo trabaja utilizando una cota superior y otra inferior, su objetivo es acortar la distancia entre ellas hasta llegar a un valor óptimo. Combina dos técnicas: una de ramificación y poda y otra de cortes. La primera es la encargada de dividir un problema grande en subproblemas más pequeños, mientras que la segunda consiste en agregar planos de corte que son restricciones lineales adicionales.
- **Algoritmo branch & price:** es una técnica utilizada para resolver problemas de optimización combinatoria como problemas de programación lineal entera, al igual que branch & cut. Este algoritmo es una combinación de otros dos métodos, branch & bound y generación de columnas. El primero se encarga de explorar el espacio de soluciones mediante un árbol y el segundo es un método encargado de resolver el subproblema de la generación de columnas que se encarga de identificar las variables más prometedoras a ser incluidas en la solución.
- **Algoritmos genéticos:** es una técnica de optimización y búsqueda inspirada en los procesos de la evolución biológica, como la mutación y herencia genética. Se utiliza para resolver problemas complejos de optimización en los que el espacio de soluciones es grande o difícil de explorar de manera exhaustiva. Trabaja simulando la evolución de una población de soluciones hacia una solución óptima. A través de generaciones, se seleccionan las mejores soluciones y se combinan para crear nuevas generaciones, esperando que las soluciones mejoren progresivamente.
- **Ant colony:** es una heurística utilizada para problemas de optimización combinatoria inspirada en el comportamiento de las hormigas al buscar rutas óptimas hacia fuentes de alimento. Este algoritmo imita las feromonas de las hormigas para recorrer el grafo. Los principales pasos son la inicialización donde se decide el nivel inicial de las feromonas, la construcción de las soluciones donde cada hormiga decide cómo recorre el grafo basándose en la feromona y alguna heurística, luego que todas las hormigas han construido sus soluciones se actualiza el valor de la feromona y por último se itera.
- **Bidirectional labeling:** es una técnica de etiquetado con el objetivo de ayudar a resolver problemas de ruteo, particularmente en situaciones donde se encuentran restricciones adicionales como incompatibilidades entre objetos. Realiza dos exploraciones de forma simultánea un etiquetado hacia delante y otro hacia atrás, cuando encuentran un nodo en común se combinan para formar una solución completa.
- **Búsqueda de vecindad adaptativa:** es una metaheurística eficaz para resolver problemas de optimización combinatoria. Su principal característica es el uso de múltiples estructuras de vecindad que cambian

sistemáticamente dentro de un procedimiento de búsqueda local. Utiliza un método de perturbación para diversificar la solución y evitar quedar atrapado en óptimos locales.

- **Camino Hamiltoniano:** es un camino que recorre un grafo tocando cada uno de sus nodos exactamente una vez, sin repetir ninguno. No necesariamente forma un ciclo, aunque si regresa al nodo inicial se convierte en un ciclo hamiltoniano.
- **Iterated local search:** es una heurística utilizada para resolver problemas complejos de búsquedas grandes. Este enfoque extiende la búsqueda local tradicional introduciendo un mecanismo para escapar óptimos locales, explorando otras áreas del espacio de soluciones mediante perturbaciones. Se debe tener en cuenta muchos valores relevantes a las instancias con las que se trabaje para encontrar el algoritmo de perturbación correcto, ya que no debe ser muy suave como para caer en un mismo mínimo local ni muy fuerte como para que cada nuevo comienzo sea aleatorio.
- **Simulated Annealing:** es una metaheurística inspirada en el proceso de enfriamiento de un metal. La idea central es comenzar con una solución aleatoria y luego ir mejorando mediante pequeñas modificaciones, aceptando soluciones peores con cierta probabilidad (la cual disminuye gradualmente simulando el proceso de enfriamiento) para evitar quedar atrapados en óptimos locales.
- **Variable Neighborhood Search:** es una metaheurística de búsqueda que se utiliza para resolver problemas de optimización combinatoria. Su idea central es cambiar sistemáticamente la estructura del vecindario de las soluciones para escapar de óptimos locales y explorar mejor el espacio de soluciones.

Revisión de la bibliografía

Diferenciamos los trabajos con problemas similares al PDPwI hallados en la literatura en función de ciertos ejes como por ejemplo si utilizan un único vehículo o no, si toman en cuenta incompatibilidades o si resuelven su problema utilizando alguna heurística.

Dentro de los que ponderamos como más relevantes, cabe aclarar que todos consideran algún tipo de incompatibilidad, para los que utilizan un único vehículo podemos destacar "*A unified branch-price-and-cut algorithm for multi-compartment pickup and delivery problems*" (Aerts-Veenstra, Cherkesly, and Gschwind 2023) donde se estudió la variante del problema one to one pickup delivery, la cual introduce que cada pedido debe ser cumplido en una ventana de tiempo y el vehículo posee múltiples compartimientos. Cada uno de estos tiene una capacidad y cierta compatibilidad con diferentes ítems. En dicho estudio se desarrolló un algoritmo de Branch & Cut, donde se utiliza la técnica de bidirectional labeling y una heurística para la aceleración en generar caminos de costo reducidos. Informaron que su algoritmo de BPC logró resolver 26,665 de 28,710 instancias en un tiempo óptimo de 3,600 segundos y pudo resolver hasta 75 request dentro de una hora.

Otro problema a mencionar que utiliza un solo vehículo es "*The family traveling salesman problem with incompatibility constraints*" (Bernardino and Paias 2021) este artículo estudió el problema family traveling salesman with incompatibility constraints (FTSP-IC), el cual generaliza al problema TSP. El objetivo es determinar un camino Hamiltoniano de mínimo costo incluyendo ciertas consideraciones: debe comenzar y terminar en un depósito, debe visitar al menos un x número de ciudades de cada "familia" y existen incompatibilidades entre "familias" que no pueden ser visitadas en la misma ruta.

Utilizaron un algoritmo de branch and cut con una heurística para conseguir límites superiores y poder eliminar algunos nodos. Se desarrollaron dos heurísticas, un algoritmo ant colony (ACO) y un algoritmo iterated local search (ILS). Al momento de realizar las pruebas utilizaron las instancias de Morán-Mirabal (Moran-Mirabal and Gonzalez-Velarde 2013) adaptadas al problema, sobre los resultados informaron que pudieron obtener valores óptimos con instancias de hasta 127 nodos con el algoritmo de branch and cut, mientras que ambas metaheurísticas lograron resolver instancias que branch and cut no pudo, además de optimizar la mayoría de las cotas superiores obtenidas por branch and cut. Para la metaheurística ACO lograron obtener un resultado promedio de 270 segundos en mayoría de las instancias y en las más grandes un promedio de 1435 segundos, mientras que utilizando mismas instancias la metaheurística logró obtener un resultado promedio de 68 segundos. Además compararon ILS y ACO utilizando instancias de las más grandes y realizando 10000 iteraciones, donde ILS tardó 2834 segundos promedio y ACO tardó 129 segundos promedio. Por último compararon las metaheurísticas entre ellas y se probó que, con misma cantidad de iteración, ILS consume menos tiempo para obtener un resultado en la mayoría de las instancias.

Por otro lado tenemos el grupo de artículos que destacamos por utilizar más de un vehículo, “*The Pickup and Delivery Problem with Time Windows and Incompatibility Constraints in Cold Chain Transportation*” (Deng et al. 2022) este artículo aborda el problema de distribución de productos perecederos bajo condiciones controladas de temperatura, considerando restricciones de incompatibilidad entre productos tanto para diferencias en los requerimientos de temperatura como de características específicas de los mismos. Además, la vida útil de los productos es un factor clave, ya que determina prioridades en la planificación.

El enfoque propuesto utiliza el algoritmo exacto branch-and-price (B&P), complementado con un algoritmo de etiquetado bidireccional asimétrico. Se trabajó con instancias propuestas por Li y Lim (Li and Lim 2001), clasificadas según la distancia entre nodos:

- Distancias cortas (horizonte de 4 horas): se resolvieron las 56 instancias propuestas, con un tiempo promedio de 52 segundos.
- Distancias medias (horizonte de 1 día): se resolvieron 58 de 63 instancias, con un tiempo promedio de 439 segundos.
- Distancias largas (horizonte de 3 días): se resolvieron 49 de 56 instancias, con un tiempo promedio de 3858 segundos.

En total, el algoritmo alcanzó soluciones óptimas en el 93% de las instancias evaluadas.

Otro artículo dentro del grupo de los que utilizan más de un vehículo aborda el problema Dial-a-Ride (DARP) “*Problema dial-a-ride: compartilhamento entre gêneros*” (Spengler, Loch, and Scarpin 2020) en un contexto de política de género, donde se propone una separación de pasajeros según su género en viajes compartidos. Utilizan una metaheurística Simulated Annealing que busca soluciones de costo mínimo para problemas combinatorios. Al momento de realizar pruebas utilizaron instancias del trabajo de Cordeau (Cordeau 2006) basado en DARP, únicamente aplicaron dos variaciones de género femenino una de 20% y otra de 40%. Cabe destacar que para las instancias de 40% la metaheurística SA no logra encontrar una solución factible, solo permitiendo un tiempo de espera de 10 minutos para los usuarios se logra de forma consistente conseguir una solución factible. En general SA mostró menos variación en relación al tamaño de la instancia, mientras que en los resultados con el trabajo de Cordeau muestra una gran variación de tiempo con instancias grandes.

Por otro lado el artículo “*Genetic Algorithm for the Pick-up and Delivery Problem with Time Window by Multi-Compartment Vehicles*” (Ousmane, Moustapha, and Adama 2021) utilizó un enfoque metaheurístico basado en algoritmos genéticos para optimizar las rutas de vehículos multi-compartimento en la recolección y entrega de productos, respetando restricciones de compatibilidad, capacidad y ventanas de tiempo. Gestionaron las incompatibilidades entre productos mediante el uso de compartimentos dedicados dentro de los vehículos, lo que permite transportar productos incompatibles en el mismo vehículo sin riesgo de conflicto.

Se realizaron pruebas con 4 instancias las cuales poseen 25, 50, 75 y 100 nodos respectivamente. En cada instancia se presentó dos soluciones, con la primera instancia y solución se realizó con un mínimo de distancia y costo total utilizando 5 autos mientras que la segunda utiliza 4 autos con mayor distancia y costo. En la segunda instancia los costos y distancia realizada por los vehículos es mayor con un menor costo a la primera solución, mientras que en la segunda solución se obtiene una pequeña distancia y un costo alto. En la tercera instancia tuvieron una distancia mínima con un coste mínimo y una distancia grande con un coste alto en la primera y segunda solución. Y por último en la cuarta instancia tuvieron un número reducido de vehículos utilizados dando un mínimo costo en la primera solución, mientras que en la segunda solución tuvieron un número alto de vehículos utilizados dando un costo un poco mayor. Utilizando sus resultados destacaron que mayor cantidad de nodos el tiempo obtenido mejora.

Por último el artículo “*Two metaheuristic approaches for solving the multi-compartment vehicle routing problem*” (Yahyaoui et al. 2020), abordó el problema de ruteo de vehículos multi-compartimento (MCVRP) donde se busca optimizar las rutas de una flota de vehículos considerando incompatibilidades entre productos, las cuales se resuelven mediante el uso de compartimentos dedicados en los vehículos. Para resolver el problema, se proponen dos enfoques metaheurísticos: un algoritmo de búsqueda de vecindad adaptativa (AVNS) y un algoritmo genético basado en cruzamiento parcialmente emparejado (GAPMX).

Para las pruebas se utilizaron 28 instancias conocidas (son una variación de las instancias de Abdulkader (Abdulkader, Gajpal, and ElMekkawy 2015)). Para las instancias pequeñas se compararon los resultados de AVNS con CPLEX (software de optimización que utiliza algoritmos exactos), dando como resultado que AVNS igualó el 100% de las soluciones óptimas alcanzadas por CPLEX. Para instancias más grandes, AVNS encontró nuevos best-known para 12 instancias, y superó los rendimientos de ACS (ant colony system) y HAC (hybridized ant colony). Para GAPMX los resultados concluyeron que encontró dos best-known para dos instancias, y superó el rendimiento de los algoritmos ACS y HAC.

Estructura de la tesis

El Capítulo 2 presenta el modelo desarrollado para abordar el Problema de Recolección y Entrega Punto a Punto con Incompatibilidades (PDPwI), incluyendo la solución propuesta en (Factorovich, Pablo. 2020) y nuestro enfoque basado en la metaheurística Variable Neighborhood Search (VNS) complementada por la heurística 3-opt. A lo largo del capítulo se detallan las decisiones de diseño, el proceso de implementación y los resultados experimentales obtenidos, junto con su correspondiente análisis. Además, se discuten las optimizaciones aplicadas sobre el tiempo de ejecución de las heurísticas mediante la imposición de límites, describiendo el procedimiento utilizado para su análisis, basado en regresiones polinomiales y comparaciones entre los resultados alcanzados.

Estas evaluaciones permiten cuantificar la mejora en tiempo y el impacto en la calidad de las soluciones respecto de la mejor solución conocida.

El Capítulo 3 expone las conclusiones generales del trabajo, así como posibles líneas de investigación futura, con énfasis en optimizaciones y mejoras potenciales para el enfoque presentado.

Finalmente, en el primer Apéndice se presentan las tablas completas con los resultados obtenidos por las heurísticas y el esquema VNS. En el segundo apéndice se incluye el código fuente desarrollado durante el proyecto.

Recolección y entrega con incompatibilidades

En este capítulo se presenta la definición formal del problema, el modelo diseñado y la implementación con un enfoque heurístico y metaheurístico. Se detallarán las razones que motivaron la elección de las heurísticas empleadas, así como los fundamentos de su integración dentro de un esquema de Variable Neighborhood Search (VNS), buscando ampliar el espacio de soluciones factibles y mejorar la calidad de las rutas obtenidas.

Definición formal del problema

Para abordar el Problema de Recolección y Entrega Punto a Punto con Incompatibilidades (PDPwI), se parte de la representación propuesta en trabajos previos, en la que la instancia se modela mediante un dígrafo $G = (P \cup D \cup \{Or, Dest\}, A)$, junto a un grafo de incompatibilidades $I = (P, E)$. En este esquema, cada vértice de P representa un punto de recolección, mientras que cada arista de E conecta pares de pedidos incompatibles, es decir, aquellos que no pueden ser transportados simultáneamente en el vehículo. Cada vértice de D representa un punto de entrega, los vértices Or y $Dest$ representan los depósitos de origen y destino respectivamente. De esta forma, las solicitudes se presentan con un vértice origen $i \in P$ y su correspondiente vértice destino $i + n \in D$. A incluirá arcos de todo vértice en $P \cup D$ a todo vértice en $P \cup D$ y viceversa.

Este formalismo permite modelar de forma explícita tanto las restricciones de precedencia entre puntos de recolección y entrega (es decir, que un nodo de entrega sólo puede visitarse después de su correspondiente nodo de recolección), como las restricciones de incompatibilidad entre ciertos bienes, derivadas de características físicas, normativas o de seguridad.

Dado que el PDPwI pertenece a la clase de problemas NP-difíciles, su resolución mediante métodos exactos implica un alto costo computacional, especialmente al aumentar la cantidad de pedidos o al considerar restricciones adicionales. Por este motivo, en este trabajo se adopta un enfoque basado en heurísticas y metaheurísticas, que permite obtener soluciones de buena calidad en tiempos razonables.

Solución propuesta

En el modelo se empleó una heurística de búsqueda local que define su vecindario mediante permutaciones controladas de los vértices de recolección y entrega dentro de la ruta, respetando las restricciones de precedencia e incompatibilidad. A esta heurística la denominamos swap a efectos de esta tesis. Esta heurística actúa modificando la secuencia de visitas sin violar las restricciones, lo que permite mejorar progresivamente una solución inicial. Su funcionamiento detallado, así como su implementación, se describe en la sección correspondiente.

Sin embargo, esta heurística presenta ciertas limitaciones al restringir los movimientos posibles dentro de la solución, lo que puede conducir a estancamientos en óptimos locales. Para superar dichas restricciones y ampliar la exploración del espacio de soluciones, se incorporó una segunda heurística, basada en 3-opt, cuyo objetivo es permitir reordenamientos más amplios en la secuencia de visitas, incrementando la capacidad de escape de mínimos locales. Entre las variantes existentes, se eligió aquella adecuada para distancias asimétricas, dado que el problema presenta este tipo de métricas. Su funcionamiento detallado se describe también en su sección específica.

Finalmente, para integrar ambas estrategias se diseñó una metaheurística de Variable Neighborhood Search (VNS), que alterna sistemáticamente los vecindarios generados por la heurística tipo swap y la 3-opt. Este enfoque permite una exploración más diversa y eficaz del espacio de soluciones, favoreciendo la superación de barreras locales y alcanzando rutas de mejor calidad en tiempos computacionales aceptables.

Heurísticas

En esta sección se presentan las heurísticas desarrolladas para generar y mejorar soluciones factibles al Problema de Recolección y Entrega Punto a Punto con Incompatibilidades (PDPwI). Estas heurísticas buscan construir rutas que respetan las restricciones de precedencia e incompatibilidad, así como mejorar iterativamente su calidad. En particular, se describen dos estrategias: una denominada swap, que modifica la secuencia de visitas mediante movimientos locales, y otra basada en el operador 3-opt, que permite reordenamientos más amplios. Ambas serán posteriormente integradas dentro de un esquema de Variable Neighborhood Search (VNS), con el fin de potenciar su capacidad de exploración.

Heurística swap

La heurística swap constituye la estrategia inicial para la mejora de rutas factibles dentro del PDPwI. Su objetivo es optimizar una solución existente modificando la secuencia de visitas, siempre respetando las restricciones de precedencia e incompatibilidad. Para ello, define su vecindario a partir de permutaciones controladas de vértices de recolección y entrega en la ruta.

El procedimiento se basa en identificar, para cada vértice de recolección, los posibles desplazamientos hacia posiciones anteriores en la secuencia, hasta encontrar un vértice de entrega de un pedido incompatible o el nodo de origen. De forma simétrica, también se exploran desplazamientos hacia adelante hasta ubicar su vértice de entrega correspondiente. Estas permutaciones garantizan la validez de las soluciones intermedias, sin necesidad de comprobar las restricciones en cada movimiento, ya que el mecanismo asegura la factibilidad en forma implícita.

Asimismo, la heurística contempla la exploración de movimientos sobre los vértices de entrega, permitiendo su desplazamiento hacia adelante hasta hallar un vértice de recolección incompatible o el no final, o hacia atrás hasta encontrar su propio vértice de recolección. Gracias a esta definición del vecindario, se logra evitar la complejidad de verificar la factibilidad de la solución generada, ya que se evita generar soluciones inválidas.

No obstante, el esquema de swap presenta ciertas limitaciones, ya que su definición restringe la amplitud de los movimientos permitidos en la ruta. En consecuencia, puede impedir alcanzar configuraciones más prometedoras dentro del espacio de soluciones. Un ejemplo de esto puede observarse en la figura 1, donde se ilustra una solución compuesta por dos pedidos incompatibles entre sí. La secuencia de visitas sigue el orden $P1 \rightarrow D1 \rightarrow P2 \rightarrow D2$, respetando tanto la precedencia como las incompatibilidades.

Figura 2.2.1.1.1: Dos pedidos incompatibles de forma continua.

En esta configuración, la heurística swap no puede mover el vértice de recolección P2 hacia una posición anterior sin violar las restricciones de incompatibilidad, lo que impide mejorar la solución, incluso cuando otras secuencias serían factibles.

En cambio, la figura 2 muestra una alternativa factible —D2 precede a P1— que podría conducir a una solución de menor costo. Esta reordenación, sin embargo, no es alcanzable por la heurística swap, ya que implicaría romper temporalmente la factibilidad de la solución durante el movimiento.

Figura 2.2.1.1.2: Dos pedidos incompatibles de forma continua (Alternativa factible).

Este tipo de situaciones motivó la incorporación de una segunda heurística, basada en 3-opt, con mayor capacidad de exploración y flexibilidad estructural. Su funcionamiento se detalla en la siguiente sección.

Heurística 3-opt

Con el objetivo de ampliar la capacidad de exploración del espacio de soluciones y superar las limitaciones observadas en la heurística swap, se incorporó una segunda estrategia basada en el operador 3-opt. Esta heurística permite realizar reordenamientos más profundos en la secuencia de visitas, favoreciendo la ruptura de bloques de pedidos que podrían generar estancamientos en óptimos locales.

El procedimiento general consiste en identificar subrutas factibles dentro de la solución inicial, entendidas como segmentos consecutivos de la ruta que respetan las restricciones de precedencia e incompatibilidad. A partir de estas subrutas, se generan todas las permutaciones posibles entre ellas (excluyendo el nodo de origen y el depósito final), buscando combinaciones que puedan mejorar la calidad de la solución. Dado que las subrutas ya cumplen las restricciones del problema, este mecanismo evita la necesidad de validaciones intermedias y permite reconstruir la ruta final únicamente tras determinar la mejor permutación alcanzada.

A efectos de esta tesis, se diseñó una versión adaptada del operador 3-opt para el PDPwI. En lugar de seleccionar aristas arbitrarias dentro de la ruta, se identifica primero cuándo el vehículo queda vacío a lo largo del recorrido. Estos momentos definen puntos de corte que permiten dividir la solución en subsoluciones o “bloques autocontenidos”, donde cada bloque incluye todos los puntos de recolección y entrega correspondientes a sus pedidos, sin solaparse con otros. De este modo, cada subsolución puede moverse libremente dentro de la ruta sin violar las restricciones del problema.

Por ejemplo, ante una solución del tipo:

Figura 2.2.1.2.1: Recorrido del vehículo donde hace pickup y delivery de cada pedido.

Puede observarse que el vehículo queda vacío tras completar D1 y nuevamente tras D2. Esto permite dividir la solución en tres bloques independientes:

- Subsolución 1: P1 → D1
- Subsolución 2: P2 → D2
- Subsolución 3: P3 → D3

Estas subrutas son completamente intercambiables entre sí, ya que respetan internamente la precedencia y no comparten elementos en simultáneo. Así, se pueden explorar todas las permutaciones posibles entre estos bloques sin necesidad de validaciones adicionales. Una de esas permutaciones podría ser:

Figura 2.2.1.2.2: Permutación del recorrido de la figura 3.

En el caso de que los tres pedidos sean mutuamente incompatibles, esta solución sería inalcanzable mediante la heurística swap, ya que implicaría violar temporalmente una restricción de incompatibilidad durante el movimiento. La heurística 3-opt, en cambio, puede reconstruir esta solución de forma válida gracias a su enfoque por bloques autocontenidos.

La figura 5 ilustra esquemáticamente el principio de operación del 3-opt, donde se eliminan tres aristas (líneas punteadas) y se generan nuevas conexiones entre los nodos (líneas continuas), evaluando las combinaciones posibles. En nuestro enfoque, las conexiones representan la reordenación de bloques enteros de tareas, en lugar de vértices individuales

Figura 2.2.1.2.3: Permutación de 3-opt específica para PDP asimétricos.

Instancias y entorno

Para evaluar el desempeño del modelo propuesto, se construyó un conjunto de instancias basado en el presentado por Dumitrescu (Dumitrescu, Irina, Stefan Ropke, Jean F. Cordeau, and Gilbert Laporte. 2008). En estas instancias, se generan $2n + 1$ puntos distribuidos aleatoriamente sobre una cuadrícula de tamaño 1000×1000 . El primer punto se utiliza como depósito, mientras que los restantes se emparejan para formar n pedidos. Cada pedido consiste en un punto de recolección i y su correspondiente punto de entrega $n+i$. El costo asociado entre dos nodos se define como la distancia euclídea (redondeada al entero más cercano).

Se generaron instancias con diferentes tamaños de problema, considerando $n = 5, 10, 15, 20, 25, 30, 35, 50, 75, 100, 200, 300, 400$ y 500 pedidos. Para cada valor de n , se construyeron 5 instancias distintas, identificadas con los sufijos a, b, c, d y e (por ejemplo, prob5a, prob5b, etc.), totalizando 125 instancias base.

Para adaptar estas instancias al problema PDP con incompatibilidades (PDPwI), se incorporó una matriz adicional que representa los niveles de incompatibilidad entre los pedidos. Así, para cada instancia base se generaron cinco variantes correspondientes a los siguientes niveles de incompatibilidad: 0%, 10%, 25%, 50% y 75%.