



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA

# **UN MÉTODO DE SOLUCIÓN EXACTO BASADO EN DESCOMPOSICIÓN PARA UN PROBLEMA DE RUTEO CON DRONES**

**SEBASTIÁN ALEJANDRO VÁSQUEZ LLORENTE**

Tesis para optar al grado de  
Magíster en Ciencias de la Ingeniería

Profesores Supervisores:  
GUSTAVO ANGULO OLIVARES  
MATHIAS KLAPP BELMAR

Santiago de Chile, Abril 2020

© MMXX, SEBASTIÁN ALEJANDRO VÁSQUEZ LLORENTE



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA

# UN MÉTODO DE SOLUCIÓN EXACTO BASADO EN DESCOMPOSICIÓN PARA UN PROBLEMA DE RUTEO CON DRONES

**SEBASTIÁN ALEJANDRO VÁSQUEZ LLORENTE**

Miembros del Comité:

GUSTAVO ANGULO OLIVARES 

MATHIAS KLAPP BELMAR 

JORGE VERA ANDREO 

EDUARDO MORENO ARAYA 

ESTEBAN SAEZ ROBERT 

Tesis para optar al grado de  
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Abril 2020

© MMXX, SEBASTIÁN ALEJANDRO VÁSQUEZ LLORENTE

*A mi familia*

## **AGRADECIMIENTOS**

En primer lugar, quiero agradecer a mi familia por ser un apoyo incondicional a lo largo de mi vida. A mi madre, Cecilia, por entregarme su amor infinito y estar ahí siempre que la necesite. A mi padre, Alejandro, por acompañarme en todas y por los sabios consejos que hasta la fecha me da. A mi hermano, Miguel, por su simpatía y constante cariño. La familia es alegre por él. A mi abuela, Inés, por conducirme a ser lo que soy y a hacer lo que hago. Es y ha sido el motor que motiva todo mi trabajo. Finalmente, a la Maca, por ser un constante apoyo en mi trabajo y en todo lo que hemos realizado juntos. Por entregarme tanto cariño y muy lindos momentos.

En segundo lugar, quiero agradecer a mis profesores. A mis supervisores, o más bien guías, Gustavo Angulo y Mathias Klapp, por su gran disposición a ayudarme y todo el conocimiento que me han entregado desde hace más de dos años. A los profesores Jorge Vera y Homero Larrain, por ser modelos de ingeniero y profesor a seguir. Muchas gracias por todos los consejos y los mensajes motivacionales.

En tercer lugar, quiero agradecer al pasillo norte de la sala de magíster del DITL, por tantas tardes (im)productivas, de conversaciones profundas y reflexivas. Mención especial a Andrés Navarrete, Carlos Céspedes, Pauli(na) Pizarro y (Vale)ria Acevedo. Sin ustedes, probablemente hubiese dejado de ir a la Universidad y todo este proceso no hubiese sido tan entretenido. También habría terminado antes, pero es un detalle que se convertirá en anécdota. Extiendo este agradecimiento a todo el DITL y a toda su gente linda de Postgrado, por los postres, los partidos y las fiestas en la salsoteca. Se extrañarán.

Por último, agradezco a la Comisión Nacional de Investigación Científica y Tecnológica por el financiamiento de uno de mis años de estudio mediante la entrega de una Beca de Magíster Nacional.

## ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	iv
ÍNDICE DE FIGURAS	vii
ÍNDICE DE TABLAS	ix
RESUMEN	x
ABSTRACT	xi
1. INTRODUCCIÓN	1
1.1. Contexto . . . . .	1
1.2. Motivación . . . . .	2
1.3. Objetivos y contribución . . . . .	3
1.4. Estructura de la tesis . . . . .	4
2. REVISIÓN BIBLIOGRÁFICA	6
2.1. Traveling Salesman Problem y extensiones . . . . .	6
2.2. Categorización de problemas de ruteo . . . . .	6
2.3. Problemas de ruteo vehicular . . . . .	7
2.4. Problemas de ruteo vehicular con <i>drone</i> . . . . .	9
2.5. Traveling Salesman Problem with Drone y extensiones . . . . .	13
2.6. Descomposición de Benders . . . . .	16
3. DEFINICIÓN DEL PROBLEMA	17
3.1. Formulación del problema . . . . .	20
4. PROPIEDADES DEL PROBLEMA	25
5. METODOLOGÍA	38
5.1. Cortes de optimalidad . . . . .	40
5.2. Cortes correctivos . . . . .	41

5.3.	Eliminación de subtours . . . . .	42
5.4.	Estructura del método . . . . .	42
6.	EXTENSIONES DEL PROBLEMA	45
6.1.	TSP-D que admite <i>loops</i> . . . . .	45
6.2.	TSP-D con restricción de autonomía de vuelo del <i>drone</i> . . . . .	47
6.3.	TSP-D con $p$ <i>drones</i> . . . . .	50
7.	RESULTADOS COMPUTACIONALES	55
7.1.	Experimentos . . . . .	55
7.2.	Análisis de resultados . . . . .	55
7.2.1.	Resultados para el TSP-D- $\ell$ . . . . .	58
7.2.2.	Resultados para el $L$ -TSP-D . . . . .	60
7.2.3.	Resultados para el $p$ TSP-D . . . . .	61
8.	CONCLUSIONES	63
	REFERENCIAS	65
	ANEXOS	71
A.	Formulación MCF para eliminación de sub-ciclos . . . . .	72
B.	Resultados computacionales para el TSP-D y el TSP-D- $\ell$ . . . . .	72

## ÍNDICE DE FIGURAS

1.1	Solución al TSP-D . . . . .	3
2.1	Ejemplo de solución al PDSTSP. Fuente: Murray & Chu (2015) . . . . .	10
2.2	Ejemplo de solución al MDRP. Fuente: Poikonen & Golden (2019) . . . . .	11
2.3	Ejemplo de solución al HDP. Fuente: Mathew et al. (2015) . . . . .	12
2.4	Solución al LSPD. Fuente: bin Othman et al. (2017) . . . . .	12
3.1	Ruta de camión y ruta de <i>drone</i> . . . . .	19
3.2	Ejemplo de soluciones óptimas al TSP y al TSP-D para una misma instancia, con $\alpha = 2$ . . . . .	21
4.1	Ejemplo ilustrativo de la Propiedad 2 . . . . .	27
4.2	Ejemplo de la aproximación propuesta . . . . .	30
4.3	Ejemplo de arcos pertenecientes a 1-atajos y 2-atajos . . . . .	32
4.4	Ejemplo de rutas en una 2-reducción . . . . .	34
6.1	Solución óptima al TSP-D- $\ell$ . . . . .	48
6.2	Solución óptima al L-TSP-D . . . . .	50
6.3	Solución al 2TSP-D . . . . .	54
7.1	Curva acumulada de soluciones óptimas al TSP-D obtenidas en el tiempo. $\alpha = 2$ . Escala logarítmica . . . . .	57
7.2	Diferencia en el número de soluciones óptimas alcanzadas entre el TSP-D y el TSP-D- $\ell$ . . . . .	59
7.3	Comparación entre soluciones para el TSP-D y para el TSP-D- $\ell$ . . . . .	60

7.4	Diferencia en el número promedio de soluciones óptimas obtenidas entre el TSP-D y el $L$ -TSP-D . . . . .	61
7.5	Diferencia en el número promedio de soluciones óptimas obtenidas entre el TSP-D y el $p$ TSP-D . . . . .	62
8.1	Ejemplo de instancia en la que resulta beneficioso poder visitar nodos con el camión. Fuente: Agatz et al. (2018) . . . . .	64

## ÍNDICE DE TABLAS

2.1	Comparación entre problemas descritos . . . . .	13
2.2	Resumen de métodos exactos disponibles en la literatura . . . . .	15
7.1	Resultados computacionales: TSP-D . . . . .	57
7.2	Resultados computacionales para el TSP-D- $\ell$ . . . . .	59
7.3	Resultados computacionales para el $L$ -TSP-D . . . . .	61
7.4	Resultados computacionales para el 2TSP-D . . . . .	62
B.1	Resultados computacionales para el TSP-D. Desviación estándar entre paréntesis . . . . .	73
B.2	Resultados computacionales para el TSP-D- $\ell$ . Desviación estándar entre paréntesis . . . . .	73

## RESUMEN

El problema del vendedor viajero con *drone* (TSP-D) es un modelo de ruteo de vehículos en el cual se debe visitar un conjunto de clientes dado con una ruta de camión que comienza y termina en un centro de distribución o por un *drone* despegado desde el camión en ruta. Este trabajo presenta una formulación compacta para modelar el TSP-D, la cual es descompuesta en dos etapas y resuelta por un algoritmo tipo Descomposición de Benders. Además, se definen distintas propiedades para fortalecer la formulación propuesta. También se explican reformulaciones para abordar distintas extensiones al TSP-D. Finalmente, a través de la resolución de una serie de instancias generadas computacionalmente, se concluye la competitividad del método propuesto frente a los distintos enfoques de solución disponibles en la literatura.

**Palabras Claves:** problema del vendedor viajero, *drones*, descomposición de Benders.

## ABSTRACT

The traveling salesperson problem with drone (TSP–D) is a vehicle routing model in which a given set of customers must be visited once, either by a truck route starting and ending at a depot or by a drone dispatched from the truck en route. This work presents a compact mixed-integer formulation for the TSP–D, which is decomposed into two stages and solved via a Benders type decomposition algorithm. Also, several properties are defined in this work to strengthen the proposed formulation. Furthermore, reformulations to address different extensions to the TSP–D are explained. Finally, numerical experiments on the TSP–D show computational competitiveness of the proposed method in comparison to solving approaches available in the literature.

**Keywords:** traveling salesman problem, drones, Benders’ decomposition.

## 1. INTRODUCCIÓN

Este capítulo introduce la utilización de *drones* en el mundo de la logística y enuncia los alcances de este trabajo. El capítulo se estructura de la siguiente forma: en la Sección 1.1 se contextualiza el uso de *drones* en operaciones logísticas; en la Sección 1.2 se define el problema a abordar; en la Sección 1.3 se presentan los objetivos y la contribución de este trabajo; y en la Sección 1.4 se describe su estructura.

### 1.1. Contexto

Empresas proveedoras de servicios logísticos internacionales como Amazon y UPS han dedicado recursos al desarrollo tecnológico de *drones* para el despacho de mercancías, buscando oportunidades para reducir costos operativos. Esto, pues un *drone* no se ve afectado por congestión vehicular y es más económico al poder ser autónomo. Si bien su capacidad es limitada, según Jeff Bezos, CEO de Amazon, aproximadamente el 86% de los paquetes que despachan no superan las 5 libras (Rose, 2013), lo que hace el envío de productos vía *drone* compatible con la realidad.

En la conferencia Amazon re:MARS del 2019, Amazon dio a conocer su última versión de un *drone* de despacho, con el cual prometen hacer despachos a domicilio de productos comprados por internet en menos de 30 minutos, aplicando tecnologías de Inteligencia Artificial avanzadas (Wilke, 2019). UPS, en conjunto con la empresa de desarrollo de *drones* Matternet, identificó el impacto del tráfico vehicular sobre la distribución de muestras médicas en Carolina del Norte, cuya demanda es urgente e implica la movilidad de objetos livianos. Frente a esto, desarrollaron un plan de despachos por *drone* el cual ha reducido los tiempos de viaje promedio de, aproximadamente, 30 minutos a 3 minutos (Adams, 2019). En 2016, UPS se unió a la ONG Gavi, the Vaccine Alliance y a la empresa de robótica Zipline para distribuir, vía *drones*, vacunas para tratar VIH/SIDA, paludismo, tuberculosis, entre otras, en zonas aledañas de Ruanda (Zaccara, 2019).

En los últimos años, empresas del rubro logístico y automotriz han desarrollado un modelo de operación combinada entre un vehículo terrestre y un *drone*. La idea es que el *drone* pueda despegar, visitar a un cliente y aterrizar en el techo del vehículo, mientras el vehículo esté en ruta. UPS ha testeado este modelo con tecnología de Workforce Group (Stewart, 2017). Otro ejemplo es Mercedes Benz, que ha desarrollado la “Vision Van”, un vehículo eléctrico que puede cargar dos *drones* en el techo (Daimler, s.f.).

Una operación combinada aprovecha las ventajas de cada modo de transporte. El *drone* es más rápido, posee mayor accesibilidad y está exento de congestión vehicular, mientras que el vehículo terrestre tiene mayor capacidad de carga y autonomía. Así, el uso conjunto brinda flexibilidad adicional para realizar despachos a clientes, lo que podría reducir costos operativos.

## 1.2. Motivación

Para operar eficientemente una operación combinada, se debe diseñar rutas especiales de despacho. El modelo básico que aborda este desafío se denomina TSP-D, el cual ha sido estudiado en la literatura científica. El TSP-D es un problema que extiende al *Traveling Salesman Problem* (TSP), cuyo objetivo es diseñar una ruta de camión que visite a todos los clientes de un grafo, una sola vez y a mínimo costo. En el TSP-D, se busca visitar a los clientes de un grafo con una ruta de camión que comienza y termina en un centro de distribución, desde ahora *depot*, o por despachos directos de un *drone* desde el camión en ruta. Además, ambos vehículos pueden moverse de manera independiente y son capaces de esperar al otro en la ubicación de los clientes dentro de la ruta. El objetivo del TSP-D es minimizar la duración total de la ruta del camión, equivalente al tiempo de llegada del camión al *depot*. La Figura 1.1 ejemplifica una solución al TSP-D sobre un grafo donde el *depot* es un nodo indexado como 1 y el resto de los nodos son clientes a visitar. El camión sigue arcos sólidos, mientras que el *drone* arcos discontinuos.

En la literatura hay análisis teórico sobre los potenciales ahorros por el uso combinado de camión y *drone*, y distintos métodos de solución.

En este trabajo se introduce un programa entero-mixto (MIP) descompuesto en dos etapas y un método de resolución exacto basado en Descomposición de Benders. El método considera la implementación de un algoritmo de *Branch-and-Cut* (B&C), para el cual se derivan cortes válidos y propiedades de una solución óptima, lo que finalmente reduce el tiempo de cómputo requerido para resolver el problema.

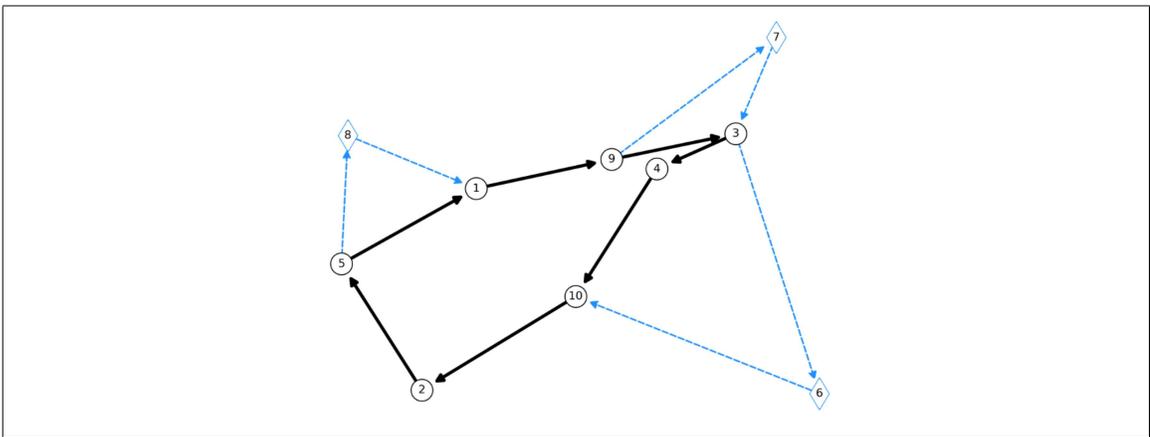


Figura 1.1. Solución al TSP-D

### 1.3. Objetivos y contribución

La pregunta de investigación de este trabajo es: ¿Existe algún beneficio al utilizar Descomposición de Benders para resolver el TSP-D?. Para responderla, se plantean los siguientes objetivos:

1. Diseñar un MIP para el TSP-D estructurado para aplicar una descomposición tipo Descomposición de Benders.
2. Implementar un algoritmo de solución mediante B&C.

3. Validar el algoritmo resolviendo una serie de instancias del TSP–D generadas computacionalmente y aplicando análisis de sensibilidad sobre ciertos parámetros del problema.

La contribución de este trabajo es el diseño de un nuevo método de solución exacto para el TSP–D que utiliza una descomposición tipo Descomposición de Benders. Esto permite aprovechar el desarrollo existente en la literatura sobre métodos de descomposición para problemas de optimización discreta. Además, la modelación propuesta permite levantar supuestos y abordar extensiones del TSP–D.

La intención de este estudio es el desarrollo y validación de un método de solución exacto para el TSP–D. En la actualidad, para resolver instancias asociadas a problemas reales, se utilizan métodos heurísticos. Por ejemplo, Moshref-Javadi et al. (2020) muestra la aplicación de un método heurístico para instancias de entre 28 a 101 clientes localizados en Sao Paulo, Brasil. Una aplicación real de un método exacto se podría obtener tras zonificar un conjunto de clientes y cubrir a los clientes de cada zona con un camión y un *drone* diferentes.

#### **1.4. Estructura de la tesis**

En el Capítulo 2 se presenta una revisión bibliográfica de la literatura relevante sobre el TSP–D y sobre extensiones al mismo. Además, se menciona la literatura relevante que estudia los métodos de resolución exacta relacionados a este trabajo.

En el Capítulo 3 se define el problema, se define la notación utilizada y se presentan los supuestos considerados al modelar. Con ello, se formula un programa de optimización entero–mixto para el TSP–D.

En el Capítulo 4 se enuncian propiedades del TSP–D que son aprovechadas en el método de solución propuesto. En el Capítulo 5 se presenta la estructura de descomposición y el algoritmo de solución. En particular, se derivan desigualdades válidas para la formulación

que son incluidos en la implementación del algoritmo, la cual también se detalla. En el Capítulo 6 se analiza cómo abordar algunas extensiones al TSP-D.

En el Capítulo 7 se describe la experimentación computacional realizada. Se presentan resultados de la aplicación del método para resolver el TSP-D, además de resultados de su aplicación sobre las extensiones consideradas.

Finalmente, en el Capítulo 8 se concluye este trabajo y se discuten futuras líneas de investigación.

## 2. REVISIÓN BIBLIOGRÁFICA

### 2.1. Traveling Salesman Problem y extensiones

El *Traveling Salesman Problem* (TSP) es un problema combinatorial ampliamente estudiado y para el cual existen distintos métodos de solución (Applegate et al., 2006). Un método para resolver el TSP es B&C, el cual combina algoritmos de inyección dinámica de planos cortantes con *Branch-and-Bound* (B&B). Este método ha resuelto a optimalidad la instancia de 85.900 nodos de la TSPLIB, la cual es una librería de instancias para el TSP utilizadas como estándar de comparación para los métodos de solución.

El *Prize Collecting Traveling Salesman Problem* (PCTSP), propuesto por Balas (1989), extiende al TSP y consiste en diseñar una ruta sobre un subconjunto de los clientes, recolectando premios por cada cliente visitado. El objetivo es maximizar la rentabilidad de la ruta (premios menos costo). Este problema se relaciona al TSP–D, pues ambos pierden premios por los clientes no visitados por el camión, pero, a veces, es rentable hacerlo. En el TSP–D, el premio perdido es igual al costo de cubrir al cliente con el *drone*. Sin embargo, en el TSP–D el premio perdido también depende del secuenciamiento de las visitas a clientes del camión.

### 2.2. Categorización de problemas de ruteo

Para comparar la dificultad de distintos problemas de ruteo, se utilizan los conceptos de sincronización propuestos por Drexl (2012), y de tareas dependientes e independientes propuestos por Otto et al. (2018).

Por un lado, los problemas de ruteo pueden exigir o no sincronización entre la flota disponible. Drexl (2012) propone cinco tipos de sincronización:

- (i) De tareas: Una tarea es una acción obligatoria que debe realizarse por uno o más vehículos. Esta sincronización hace referencia a la necesidad de asignar tareas a una flota.
- (ii) De operación: Las operaciones son acciones que deben ejecutarse en ruta para poder realizar tareas. Esta sincronización hace referencia a la necesidad de decidir cuándo y dónde se realizan las operaciones.
- (iii) De movimiento: Hace referencia a los casos en que se poseen vehículos autónomos y no autónomos, y uno requiere la presencia del otro para moverse. Este requerimiento puede imponerse en algún punto de la ruta o específicamente en el *depot*.
- (iv) De carga: Hace referencia a los casos en que la carga de los vehículos debe ser considerada al momento de realizar una operación.
- (v) De recurso: Hace referencia a los casos en que existen recursos comunes para realizar tareas y es escaso.

Por otro lado, según Otto et al. (2018), los problemas de ruteo exigen el cumplimiento de tareas. Los problemas con tareas independientes son aquellos en que la flota tiene tareas asignadas y la realización de una no depende de la realización de las demás. Los problemas con tareas dependientes son aquellos en que la tarea asignada a un vehículo depende de la realización de otra tarea asignada a otro vehículo. Los problemas con tareas independientes poseen una estructura propicia para ser descompuesta por vehículo, es decir, conociendo la asignación de tareas, el problema de cómo realizarlas es de cada vehículo y se puede resolver de manera individual. Esta propiedad no se conserva en problemas con tareas dependientes.

### **2.3. Problemas de ruteo vehicular**

A continuación, se presentan extensiones al TSP que consideran múltiples vehículos.

En el *Truck-and-Trailer Routing Problem* (TTRP), propuesto por Chao (2002), se planifica la visita a clientes con un camión y un *trailer* acoplado, ambos con capacidad para llevar carga. Existen clientes que solo pueden ser visitados por el camión, por lo que en partes de la ruta este debe desacoplar el *trailer*, visitar dichos clientes y, posteriormente, volver para acoplarlo. Además, se cuenta con la posibilidad de traspasar carga del *trailer* al camión.

Según la categorización de Drexl (2012), este problema exige los cinco tipos de sincronización. En particular, exige sincronización de movimiento en ruta porque se poseen dos vehículos capaces de separarse en ruta pero que luego deben reencontrarse. A diferencia del TTRP, el TSP-D tiene la complejidad adicional de que el *drone* es capaz de moverse y visitar nodos por sí solo mientras se encuentra separado del camión. Nótese que ambos problemas exigen sincronización de carga, pues en el TSP-D el camión traspasa carga al *drone* para que pueda realizar tareas, mientras que en el TTRP lo mismo sucede entre el *trailer* y el camión.

El TTRP no podría clasificarse según Otto et al. (2018), pues solo uno de los vehículos es capaz de realizar tareas.

Otra generalización al TSP es el *Vehicle Routing Problem* (VRP), propuesto por G. B. Dantzig & Ramser (1959). Este problema considera una flota de vehículos y busca encontrar rutas para cada uno de ellos, de forma que se visite un conjunto de clientes a mínimo costo. Existen diversos métodos de solución y extensiones al VRP; ver Toth & Vigo (2002).

Según la categorización de Drexl (2012), este problema exige sincronización de tareas y de movimiento, pero solo en el *depot*, pues la única exigencia común para los vehículos es realizar una ruta que comience y termine en el *depot*. Por otro lado, el TSP-D exige sincronización en ruta, lo que implica que el *drone* se puede encontrar con el camión en más de un punto de la ruta. Según la categorización de Otto et al. (2018), el VRP exige el cumplimiento de tareas independientes.

## 2.4. Problemas de ruteo vehicular con *drone*

A continuación, se presentan problemas de diseño de rutas de visita a clientes para vehículos asistidos por *drones*. Para más detalles, revisar de Freitas & Penna (2020).

Murray & Chu (2015) proponen el *Parallel Drone Scheduling Traveling Salesman Problem* (PDSTSP), donde un camión y una flota de *drones* visitan clientes de forma independiente. El camión ejecuta una ruta que visita un subconjunto de clientes, mientras que el resto es visitado por *drones* despachados desde el *depot* y realizando viajes de ida y vuelta, restringidos por su autonomía. Un ejemplo de solución se presenta en la Figura 2.1, donde solo los clientes representados por nodos cuadrados son accesibles por el *drone*, y donde el camión sigue arcos sólidos, mientras que el *drone* arcos punteados. Ulmer & Thomas (2018) presenta el *Same-Day Delivery Problem with a Heterogeneous Fleet of Drones and Vehicles*, el cual extiende al PDSTSP al considerar un problema dinámico-estocástico donde los pedidos se reciben dinámicamente en el tiempo.

Según la categorización de Drexler (2012), el PDSTSP exige sincronización en movimiento, pero solo en el *depot*. Según la categorización de Otto et al. (2018), exige realizar tareas independientes. A diferencia del PDSTSP, el TSP-D exige que el camión y el *drone* estén sincronizados en ruta, además de exigir la realización de tareas dependientes.

El *Carrier-Vehicle Travelling Salesman Problem* (CVTSP) (Garone et al., 2008, 2011, 2014; Gambella et al., 2017) es un problema en el que se planifica la operación de un vehículo transportador (*carrier*) y un vehículo transportado de mayor velocidad y con autonomía limitada para visitar clientes. En este caso, solo el vehículo transportado puede visitar clientes, siendo desplegado desde el *carrier* y volviendo a él luego de haber realizado la visita. La función del *carrier* es acercar al vehículo transportado a los clientes a visitar.

Poikonen & Golden (2019) proponen el *Mothership and Drone Routing Problem* (MDRP), el cual estudia el CVTSP en el contexto de un *drone* como vehículo autónomo transportado. Un ejemplo de solución se presenta en la Figura 2.2, donde el *carrier* sigue

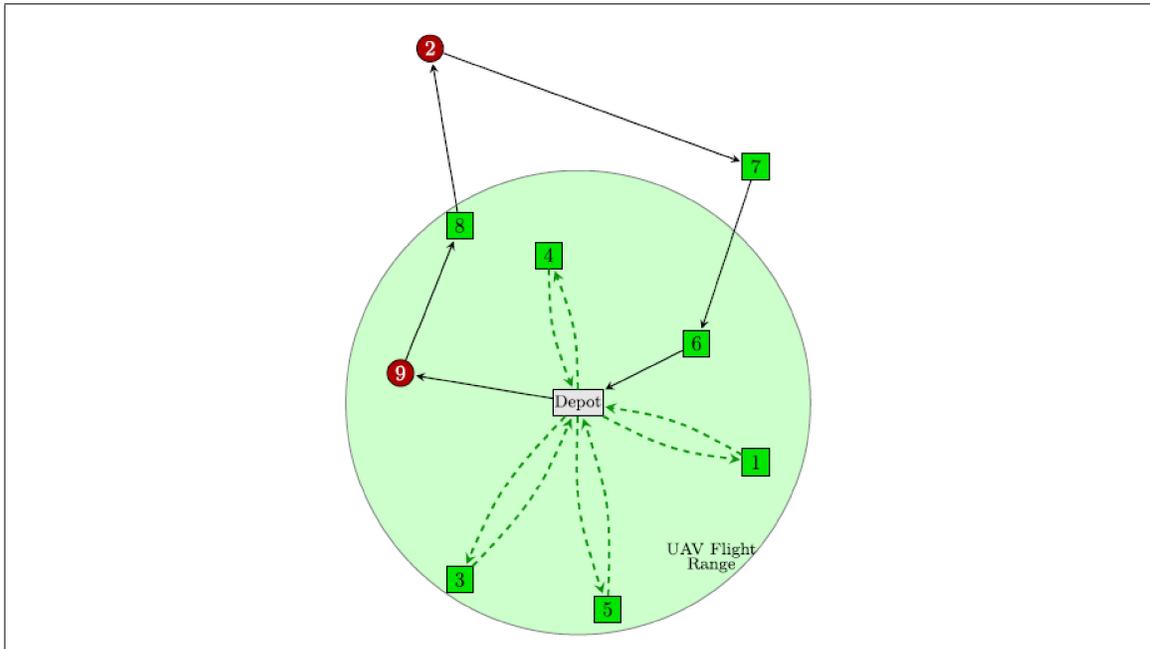


Figura 2.1. Ejemplo de solución al PDSTSP. Fuente: Murray & Chu (2015)

arcos negros y el vehículo transportado arcos rojos. Los círculos rojos son puntos en el espacio y en la ruta del *carrier* en los cuales se despliega o recupera el vehículo transportado, que visita los círculos azules. El cuadrado representa la ubicación del *depot*.

A diferencia del TSP-D, en el MDRP el *carrier* puede desplegar y recuperar al *drone* en cualquier punto del espacio. Por otro lado, en el TSP-D el *carrier* también visita a clientes.

El *Heterogeneous Delivery Problem* (HDP) considera un camión con un *drone* a bordo. Existen clientes que deben ser visitados de forma obligatoria por el *drone* y puntos en los cuales se pueden realizar despachos del *drone*. El *drone* puede visitar clientes mientras el camión sigue en ruta. El objetivo del HDP es diseñar rutas de camión y *drone* que minimicen los costos totales de ruteo. Además, se consideran tres costos de arco: uno para el vuelo de *drone*, uno para el camión vacío y otro para el camión portando el *drone*.

Un ejemplo de solución al HDP se presenta en la Figura 2.3, donde los nodos  $d$  son de visita obligatoria, mientras que los nodos  $w$  son de despliegue/aterrizaje y  $w_0$  es del *depot*.

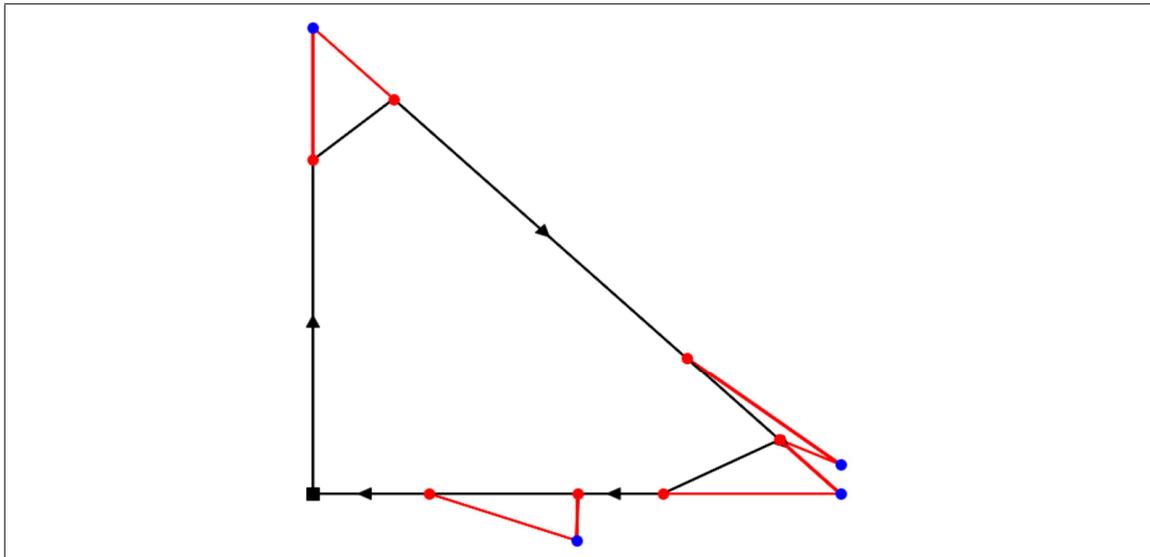


Figura 2.2. Ejemplo de solución al MDRP. Fuente: Poikonen & Golden (2019)

El camión sigue arcos sólidos, el *drone* arcos punteados y los arcos remarcados son los escogidos en la solución. Los arcos azules representan viajes del camión sin el *drone* a bordo, mientras que los negros representan viajes del camión junto al *drone*. Mathew et al. (2015) presenta este problema y propone una reducción del mismo al *Generalized TSP* (GTSP) (Tsiligirides, 1984; Golden et al., 1987), el cual se resuelve utilizando métodos de solución exactos para el GTSP.

A diferencia del TSP-D, el HDP no considera la posibilidad de que el camión visite clientes y considera los puntos de despegue o aterrizaje del *drone* como un subconjunto predefinido de todos los puntos a visitar.

bin Othman et al. (2017) presenta el *Last-Stretch Delivery Problem* (LSDP), que considera un camión con un *drone* a bordo y se diseñan rutas de *drone* para visitar clientes a mínimo costo, dada una ruta del camión. En el mismo trabajo se muestra que, para el caso en que el camión visita exactamente un cliente más que el *drone* y no es capaz de esperarlo en el punto donde lo desplegó, existe un método de solución para el LSDP en tiempo polinomial. En la Figura 2.4 se muestra un ejemplo, donde dada una ruta del camión

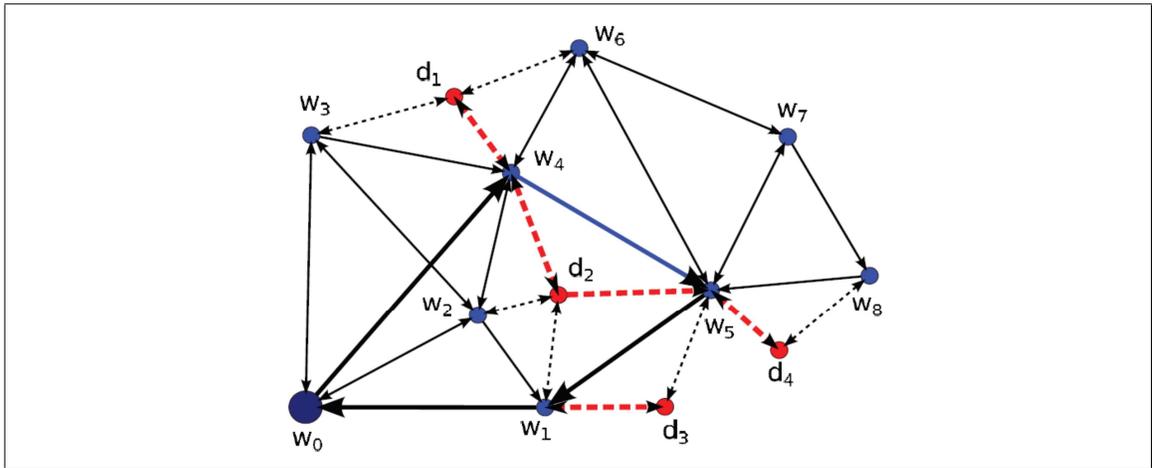


Figura 2.3. Ejemplo de solución al HDP. Fuente: Mathew et al. (2015)

(Figura 2.4a), se resuelve el LSPD (Figura 2.4b). Los arcos remarcados en la Figura 2.4b son escogidos para formar la ruta del *drone* en la solución.

La solución LSPD representa una solución parcial al TSP-D, ya que no se decide la ruta del camión. Por ende, el TSP-D es más complejo.

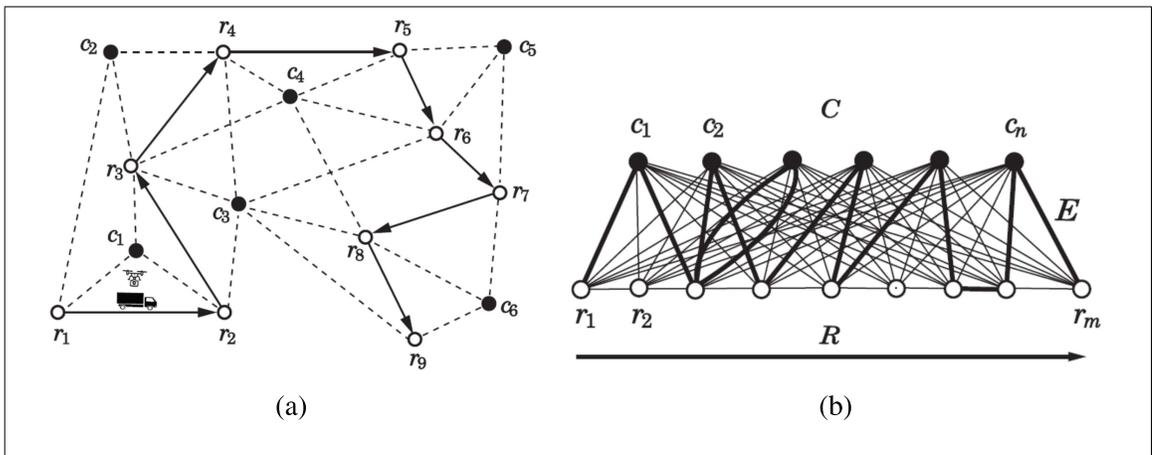


Figura 2.4. Solución al LSPD. Fuente: bin Othman et al. (2017)

En la Tabla 2.1 se presenta una comparación entre los distintos problemas presentados previamente que consideren un camión y uno o más *drones*, en función de las categorías presentados por Drexel (2012) y Otto et al. (2018). Para efectos comparativos, se considera el

VRP con un camión y un *drone*. Se agrega, además, el concepto de sincronización temporal. Este considera los casos en que ambos vehículos son capaces de moverse separados y de manera simultánea. Se considera que el LSDP no exige sincronización temporal, pues la ruta del camión se define previamente.

Tabla 2.1. Comparación entre problemas descritos

Problema	Tareas		Sincronización de movimiento		Sincronización de operación	Sincronización de carga	Sincronización temporal
	Independiente	Dependientes	En depot	En ruta			
TTRP		✓*		✓	✓	✓	
VRP	✓		✓				✓
PDSTSP	✓		✓				✓
CVTSP		✓*		✓	✓		✓
HDP		✓*		✓	✓		✓
LSDP		✓*		✓	✓		
TSP-D		✓		✓	✓	✓	✓

\*Uno de los vehículos se mueve, pero no realiza tareas

## 2.5. Traveling Salesman Problem with Drone y extensiones

El TSP–D fue propuesto por Murray & Chu (2015) y desde la fecha se han propuesto diversos métodos de solución. Por un lado, en la literatura existen métodos heurísticos o aproximados para el TSP–D. Entre ellos, se encuentra el trabajo de Agatz et al. (2018), el cual se presenta un modelo entero–mixto y propone heurísticas del tipo *route first-cluster second*. En una primera etapa se visitan todos los clientes y en la segunda se particiona la solución en una ruta del camión y una del *drone*. Para ello, se utilizan técnicas de búsqueda local y programación dinámica. De la misma forma, Ha et al. (2018) propone un método *route first-cluster second* con la salvedad de que asume que los puntos de despegue y aterrizaje del *drone* en un operación deben ser distintos. Distintas técnicas se han abordado en métodos heurísticos para resolver el TSP–D, como algoritmos genéticos (Ferrandez et al., 2016), *simulated annealing* (Ponza, 2016) y *variable neighborhood search* (de Freitas & Penna, 2020). Para mayor detalle sobre estos, revisar Otto et al. (2018).

Por otro lado, se pueden encontrar métodos exactos para el TSP–D. La resolución directa de la formulación entera de Agatz et al. (2018) es posible para instancias de hasta diez (10) clientes. Bouman et al. (2018) propone un método basado en Programación

Dinámica (DP) capaz de resolver instancias de hasta dieciséis (16) clientes. Este método se compone de tres pasos. Primero, la generación de una tabla de costos de ruteo de camión, segundo, la generación de una tabla de costos asociada a paralelizar la operación con el *drone* y, finalmente, la obtención recursiva de la asignación óptima para el *drone* en la ruta del camión. Para esta última etapa se propone como extensión un algoritmo de búsqueda  $A^*$ , con el cual se obtienen, en ciertos casos, mejores tiempos de resolución. Este método resuelve el TSP-D en tiempo  $O(3^n n^2)$ .

Van Dijk et al. (2018) propone un método basado en B&C, utilizando la formulación propuesta por Agatz et al. (2018). En él, se derivan cortes propios del TSP para el TSP-D, los cuales se agregan dinámicamente a la formulación utilizada. En el problema que aborda, se asume que el camión puede visitar un número máximo de clientes entre un despliegue del *drone* y su posterior aterrizaje en el camión. Yurek & Ozmutlu (2018) propone un método iterativo basado en descomposición. Esta descomposición consiste en dos etapas, en donde en la primera se rutea el camión y en la segunda se rutea el *drone* dada la ruta del camión escogida. Se generan todas las posibles rutas de camión y se ordenan de la más barata a la más cara, según su evaluación en la función objetivo. Con un programa entero-mixto, se obtiene la ruta de *drone* óptima asociada a cada ruta de camión, de forma secuencial, y se termina el algoritmo cuando se encuentra una ruta de camión más cara que la cota superior (mejor solución encontrada). Las instancias resueltas a optimalidad alcanzan tamaños de hasta doce (12) clientes.

Tang et al. (2019) propone una formulación basada en programación por restricciones, con la cual es capaz de resolver instancias de hasta dieciocho (18) clientes. Dell'Amico et al. (2019) propone una modificación a la formulación propuesta por Murray & Chu (2015), la cual es cuadrática en cuanto a sus variables. Además, se realizan pruebas experimentales que muestran una mayor fortaleza con respecto a la formulación original. Finalmente, Roberti & Ruthmair (2019) propone un MIP y un método de resolución basado en DP embebido en *Branch-and-Price* (B&P), con el cual también aborda distintas extensiones

al TSP–D. Además, se propone una relajación del DP tipo *ng-route*, una estrategia de estabilización dual y una estrategia de ramificación.

La Tabla 2.2 resume los métodos exactos de solución al TSP–D disponibles en la literatura.

Una extensión natural al TSP–D es el *Vehicle Routing Problem with Drone* (VRP–D) el cual considera una flota de camiones y una de *drones*. Poikonen et al. (2017) realiza un estudio del tipo *Worst-case analysis*, del cual deriva cotas teóricas para el ahorro y compara así distintas configuraciones de flota y velocidad. Wang & Sheu (2019) propone un método basado en *Branch-and-Price* para el problema asumiendo un conjunto predeterminado de puntos de despliegue y/o aterrizaje y permitiendo la flexibilidad de intercambiar *drones* entre los camiones en ruta. Pugliese & Guerriero (2017) propone un programa entero-mixto para modelar el VRP–D con ventanas horarias, con el cual resuelve instancias de hasta diez (10) clientes.

Otra posible extensión es la propuesta por Carlsson & Song (2017), denominada *Horsefly Routing Problem*, en la cual el *drone* puede despegar y aterrizar en cualquier punto de la ruta del camión. Se realiza análisis teórico sobre el plano euclidiano y simulación numérica en tiempo real, para finalmente derivar resultados que cuantifican el ahorro por el uso combinado de camión y *drone*.

Tabla 2.2. Resumen de métodos exactos disponibles en la literatura

Autores	Métodos
Murray & Chu (2015); Agatz et al. (2018)	MIP
Bouman et al. (2018)	DP ( $O(3^n n^2)$ )
Van Dijk et al. (2018)	B&C
Yurek & Ozmutlu (2018)	Enumeración (MIP para ruteo de <i>drone</i> )
Tang et al. (2019)	Programación por restricciones
Dell’Amico et al. (2019)	MIP
Roberti & Ruthmair (2019)	DP (Embebido en B&P, <i>ng-route relaxation</i> )
Este trabajo	Descomposición tipo Benders (B&C)

## 2.6. Descomposición de Benders

El método lineal de Descomposición de Benders, propuesto por Benders (1962), consiste en el fraccionamiento de un problema en dos más pequeños, un problema maestro y un subproblema, los cuales se resuelven iterativamente para encontrar la solución óptima al problema original. La solución al problema maestro sirve como parámetro para encontrar una solución al subproblema. A su vez, la solución al subproblema permite generar un corte para el problema maestro que reduce su espacio de búsqueda. Así, la idea detrás de este método es acotar el dominio del problema maestro iterativamente hasta encontrar la solución óptima al problema original. Adicionalmente, permite simplificar problemas complejos mediante la fijación de variables que evitan la descomposición directa del problema.

Este método ha sido ampliamente utilizado en diversos problemas combinatoriales y de gran tamaño, con estructura apropiada. Algunas áreas que lo han aprovechado son Planificación, Gestión de sistemas de salud, Sistemas de transporte y Sistemas de energía, entre otras. Para mayor detalle del método y sus aplicaciones, ver Rahmaniani et al. (2017). Algunas mejoras computacionales son presentadas en Angulo et al. (2016).

Para problemas descomponibles en dos etapas, donde las variables del maestro son binarias y las del subproblema enteras-mixtas, Laporte & Louveaux (1993) propone un método alternativo a la Descomposición de Benders denominado *Integer L-Shaped Method*. En este, los cortes de optimalidad son exactos para la solución del maestro pero redundantes para otras soluciones, es decir, son puntuales. Este método ha sido utilizado para resolver, por ejemplo, problemas de ruteo bajo incertidumbre (Laporte & Louveaux, 1998; Laporte et al., 2002). Otras aplicaciones y extensiones del método se pueden apreciar en Sahinidis (2004).

### 3. DEFINICIÓN DEL PROBLEMA

En este capítulo, se define el TSP–D, su notación y supuestos. Además, se propone un MIP que lo modela.

El TSP–D se define sobre un grafo dirigido  $G = (N, E)$ , donde  $N$  es el conjunto de nodos y  $E$  el conjunto de arcos. El conjunto de nodos está indexado de 1 a  $n$ , donde el nodo 1 es un *depot* desde el cual se despacha el servicio de visitas. Los conjuntos  $\delta^+(i), \delta^-(i)$  contienen los arcos salientes y entrantes al nodo  $i$ , respectivamente. Se definen los tiempos de viaje en cada arco  $c, d : N^2 \rightarrow \mathbb{R}_+$  para el camión y el *drone*, respectivamente. La relación entre los tiempos se asume lineal y constante, es decir,  $c = \alpha \cdot d$ , con  $\alpha \geq 1$ .

Para representar cada despacho de un *drone*, se define una *operación*. Presentada por Agatz et al. (2018), una operación consiste en un despegue del *drone* desde el camión en un nodo  $i \in N$ , una posterior visita a un nodo  $k \in N$  y un aterrizaje sobre el camión en el nodo  $j \in N$ , donde  $i \neq k, i \neq j, k \neq j$ . Considerando lo anterior, se define el conjunto  $O$  de operaciones factibles como

$$O = \{(i, k, j) \in N^3 : i \neq k, i \neq j, k \neq j, k \neq 1\} \cup \{(1, k, 1) : k \in N \setminus \{1\}\}.$$

Adicionalmente, se definen los conjuntos  $\delta_O^+(i)$  y  $\delta_O^-(i)$  que contienen todas las operaciones factibles que comienzan y terminan en el nodo  $i$ , respectivamente.

Sea  $\mathcal{R}$  una ruta del camión expresada como una secuencia de nodos, *i.e.*,  $\mathcal{R} = \{r_0, r_1, \dots, r_{|\mathcal{R}|-1}\}$ , donde  $r_0$  es el *depot*. La duración de una operación  $(i, k, j)$  ejecutada, dada una ruta de camión  $\mathcal{R}$ , se define como el valor máximo entre el tiempo de viaje del camión en la sección de la ruta  $\mathcal{R}$  entre  $i$  y  $j$ , denotado  $\hat{c}_{ij}(\mathcal{R})$ , y el tiempo del despacho directo del *drone* al nodo  $k$ , dado por  $d_{ik} + d_{kj}$ . En otras palabras, la duración de una operación ejecutada corresponde al tiempo de viaje del camión en la sección de la ruta  $\mathcal{R}$  entre  $i$  y  $j$  más un retraso  $w_{ikj}(\mathcal{R})$  por esperar al *drone* en el nodo  $j$ , dado por

$$w_{ikj}(\mathcal{R}) = (d_{ik} + d_{kj} - \hat{c}_{ij}(\mathcal{R}))_+ \triangleq \max\{d_{ik} + d_{kj} - \hat{c}_{ij}(\mathcal{R}), 0\}. \quad (3.1)$$

La Figura 3.1a ejemplifica una ruta del camión  $\mathcal{R}$  para la cual  $r_2 = i$ ,  $r_6 = j$  y el *depot* aparece como nodo cuadrado. En la Figura 3.1b se ilustra la operación  $(i, k, j)$ , donde el viaje del camión se representa por arcos sólidos y el del *drone* por arcos punteados.

El TSP-D consiste en diseñar una ruta del camión que despliega un *drone* en ruta, visitando a todos los clientes con, al menos, uno de los vehículos. Su objetivo es que dicha ruta del camión sea de mínima duración total, esto es, la duración de la ruta del camión más la espera del mismo debido a la ruta del *drone*.

A continuación se resumen los supuestos utilizados.

- (i) El *drone* es, al menos, tan rápido como el camión, *i.e.*  $\alpha \geq 1$ .
- (ii) Los tiempos  $c$  y  $d$  definen una métrica sobre  $N^2$ .
- (iii) Una operación debe empezar y terminar en nodos distintos del grafo.
- (iv) El *drone* puede visitar solo un nodo por operación.
- (v) Una vez aterrizado el *drone*, este puede despegar inmediatamente.
- (vi) Todo nodo puede ser visitado por el camión o el *drone*.
- (vii) El *drone* posee suficiente autonomía para realizar cualquier operación factible.
- (viii) Los nodos pueden ser visitados una sola vez por el camión.
- (ix) Ambos vehículos pueden esperar en ruta.
- (x) El tiempo de atención es despreciable en comparación con el tiempo de viaje.

(i) se puede considerar por el hecho de que el *drone* es más liviano y no se ve afectado por congestión. (ii) implica que se conserva la propiedad de desigualdad triangular en los tiempos, pero no se consideran efectos como la congestión en el ruteo del camión. Además, los tiempos del camión y del *drone* definen la misma métrica. (iii) se justifica con el hecho de que la tecnología actualmente desarrollada presenta dificultades para aterrizar o despegar un *drone* desde el camión en movimiento Orphanides (2016), por lo que el camión debe estar prácticamente detenido para comenzar o finalizar una operación. Es posible modificar el modelo propuesto para levantar este supuesto. El detalle se encuentra en el Capítulo 6. Con respecto a (iv), para las grandes empresas de *e-commerce*, como Amazon, la mayoría

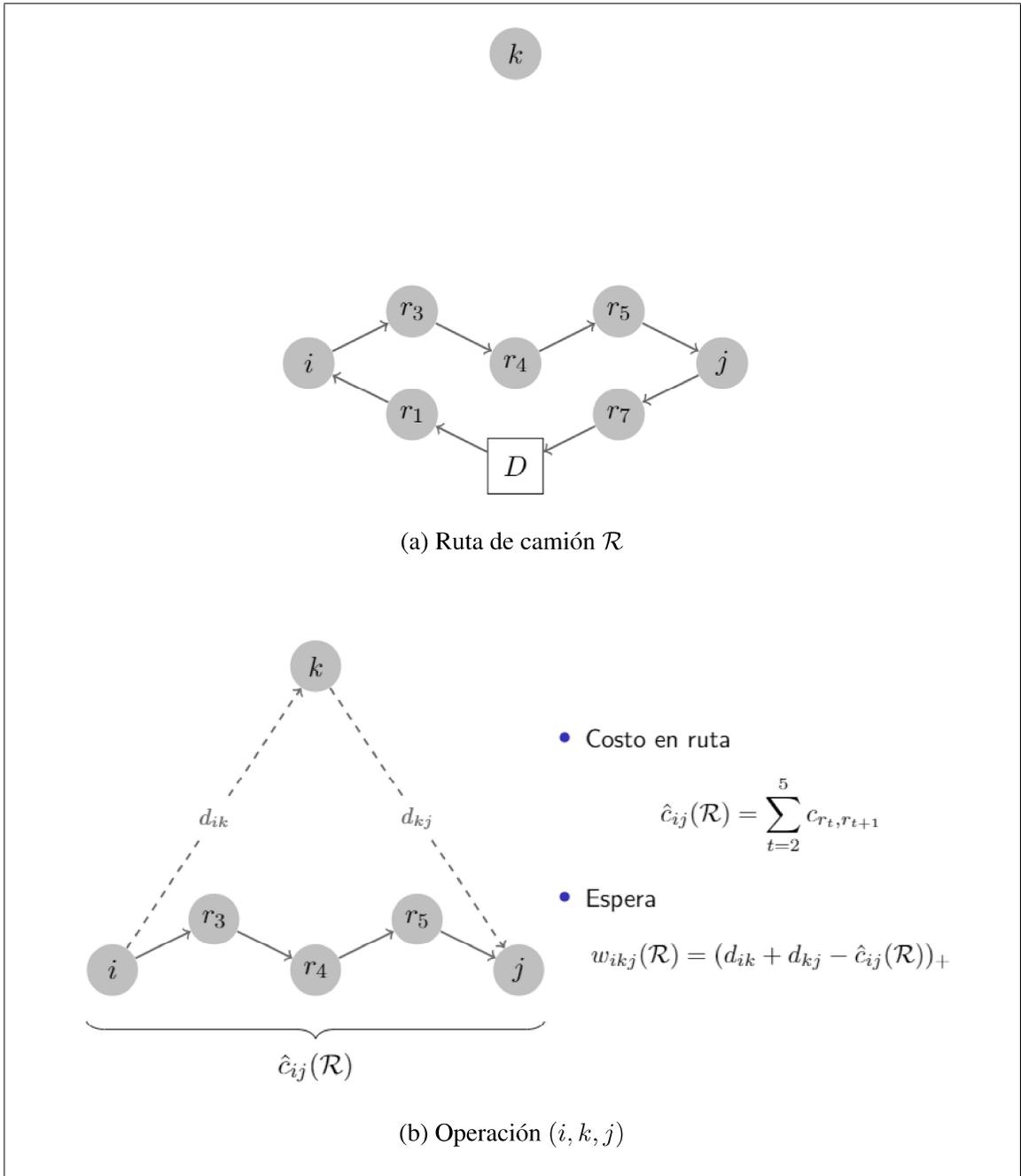


Figura 3.1. Ruta de camión y ruta de *drone*

de sus despachos no superan las 5 libras Rose (2013). Así, se considera que los despachos a utilizar en este trabajo son de 5 libras, equivalente a la capacidad de un *drone*. (v) puede ser aplicado si se considera la posibilidad de intercambiar la batería utilizada del *drone* por

una cargada cuando aterrice, o que el camión porte dos *drones* y los utilice de a uno a la vez. En cuanto a (ix), la espera del *drone* en ruta se puede considerar como una operación más lenta en la cual no espera, sino que coordina de manera exacta el tiempo en el cual se encuentra con el camión. Finalmente, por simplicidad, se asume (x), pero el tiempo de atención puede ser incorporado como tiempo adicional al tiempo de viaje, como lo hace Dell'Amico et al. (2019).

Un ejemplo de solución óptima al TSP–D se ilustra en la Figura 3.2. En la misma, se presenta una comparación con la solución al TSP, ambos aplicados a una instancia de 8 nodos, donde  $\alpha = 2$ . En este caso, el TSP–D produce un ahorro de tiempo de  $(12.48 - 6.92)/12.48 = 44\%$  sobre la solución óptima al TSP.

### 3.1. Formulación del problema

A continuación, se modela el TSP–D mediante un MIP. Para modelar la ruta de camión, se define la variable binaria  $x_e \in \{0, 1\}$ , que es igual a 1 si y solo si el camión utiliza el arco  $e = (i, j) \in E$ . También se define la variable binaria  $\gamma_i \in \{0, 1\}$ , que es igual a 1 si y solo si el camión visita el nodo  $i \in N$ . Luego, se define el conjunto  $\mathbb{X}$  de rutas de camión factibles como todo vector  $(x, \gamma)$  que cumple con

$$\sum_{e \in \delta^+(i)} x_e = \sum_{e \in \delta^-(i)} x_e = \gamma_i \quad i \in N \quad (3.2a)$$

$$\sum_{e \in \delta^+(S)} x_e \geq \max_{i \in S} \{\gamma_i\} \quad \emptyset \neq S \subset N : 1 \notin S \quad (3.2b)$$

$$x_{1j} \leq \sum_{(i,1) \in \delta^-(1): i \leq j} x_{i1} \quad (1, j) \in \delta^+(1) \quad (3.2c)$$

$$x_{i1} \leq \sum_{(1,j) \in \delta^+(1): i \leq j} x_{1,j} \quad (i, 1) \in \delta^-(1) \quad (3.2d)$$

$$\sum_{i \in N} \gamma_i \geq \left\lceil \frac{n}{2} \right\rceil \quad (3.2e)$$

$$\gamma_1 = 1 \quad (3.2f)$$

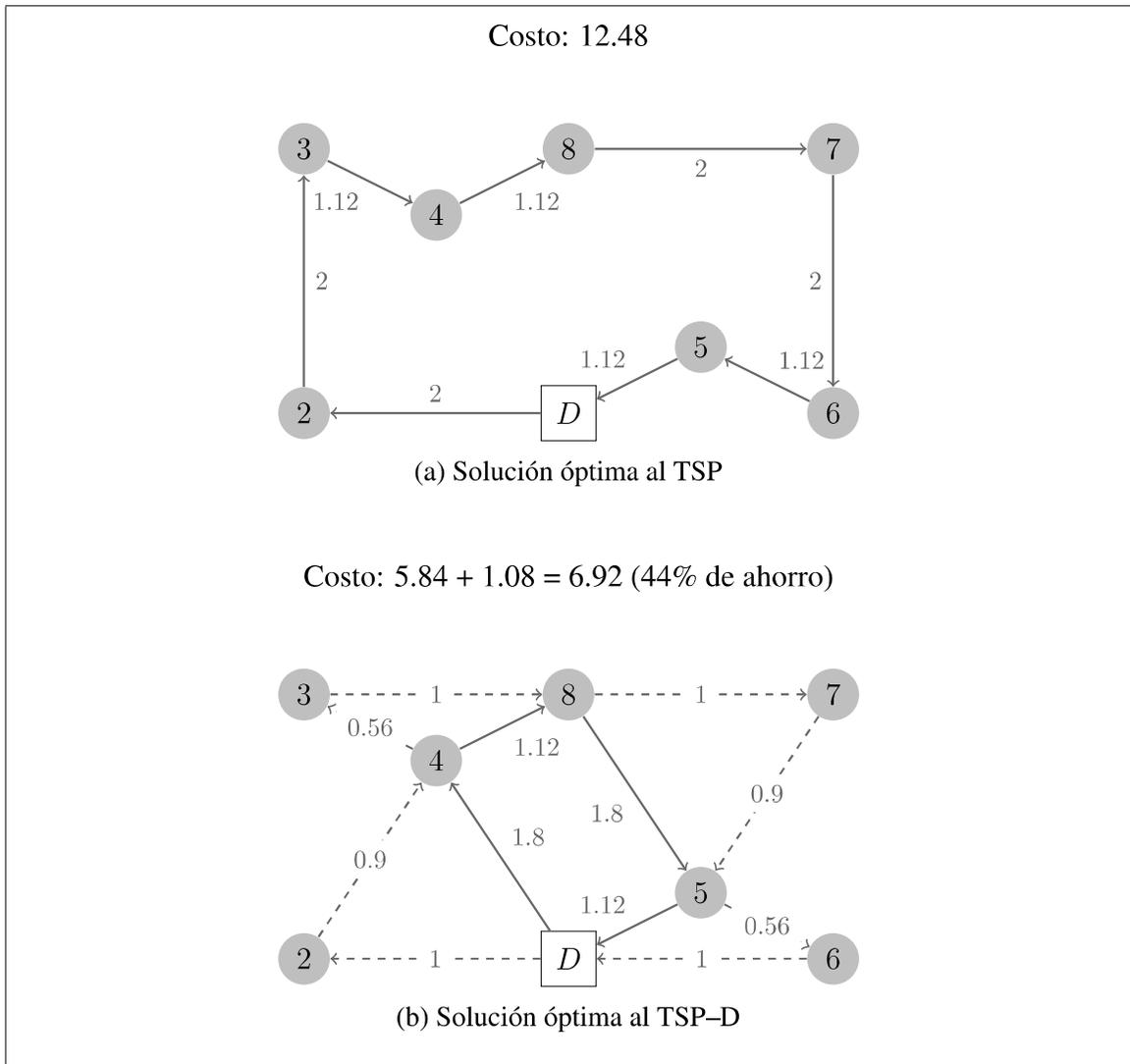


Figura 3.2. Ejemplo de soluciones óptimas al TSP y al TSP-D para una misma instancia, con  $\alpha = 2$

$$x \in \{0, 1\}^{|E|} \tag{3.2g}$$

$$\gamma \in \{0, 1\}^{|N|} . \tag{3.2h}$$

La familia de restricciones (3.2a) fuerza a que un nodo visitado deba tener un arco entrante y otro saliente, y (3.2b) garantiza que la solución no tendrá sub-ciclos. La familia (3.2c)–(3.2d) elimina soluciones simétricas, exigiendo que el nodo posterior al *depot* tenga

un índice menor al nodo anterior. (3.2e) fuerza a que el camión visite, al menos, la mitad de los nodos del grafo. Si el camión visita menos nodos, entonces no existirá una ruta de *drone* factible. (3.2f) fuerza a que el *depot* sea visitado por el camión. Finalmente, (3.2g) y (3.2h) definen la naturaleza de las variables. Por simplicidad, hablaremos de un valor específico del vector  $(x, \gamma)$  como una ruta del camión.

Dada una ruta del camión  $(x, \gamma)$ , se define el conjunto  $\mathbb{Y}(x)$  de rutas del *drone* factibles. Para construir  $\mathbb{Y}(x)$ , se definen las variables binarias  $f_e \in \{0, 1\}$ , la cual es igual a 1 si y solo si el *drone* viaja a lo largo del arco  $e = (i, j) \in E$  a bordo del camión, y  $o_r \in \{0, 1\}$ , la cual es igual a 1 si y solo si se ejecuta la operación  $r = (i, k, j) \in O$ . También se define la variable continua  $w_{ikj} \in \mathbb{R}_+$ , que cuantifica la espera del camión asociada a la operación  $(i, k, j) \in O$ .

Así,  $\mathbb{Y}(x)$  corresponde al conjunto de vectores  $(f, o, w)$  para los cuales existe  $(\gamma, y, u, v, \hat{c})$  tal que

$$\gamma_i = \sum_{e \in \delta^+(i)} x_e \quad i \in N \quad (3.3a)$$

$$u_i - u_j + 1 \leq n \cdot (1 - x_{ij}) \quad (i, j) \in E : j \neq 1 \quad (3.3b)$$

$$u_1 = 1 \quad (3.3c)$$

$$2\gamma_i \leq u_i \leq n \cdot \gamma_i \quad i \in N \setminus \{1\} \quad (3.3d)$$

$$\sum_{i \in N \setminus \{j\}} y_{ij} = u_j - 1 \quad j \in N \quad (3.3e)$$

$$y_{ij} + y_{ji} \leq \min\{\gamma_i, \gamma_j\} \quad i, j \in N : i \neq j \quad (3.3f)$$

$$\gamma_i + \gamma_j - 1 \leq y_{ij} + y_{ji} \quad i, j \in N : i \neq j \quad (3.3g)$$

---


$$v_i - v_j + c_{ij} \leq M \cdot (1 - x_{ij}) \quad (i, j) \in E : j \neq 1 \quad (3.3h)$$

$$v_i \leq \sum_{e \in E} x_e c_e - \sum_{e \in \delta^-(1)} x_e c_e \quad i \in N \setminus \{1\} \quad (3.3i)$$

$$v_1 = 0 \quad (3.3j)$$

$$\hat{c}_{ij} \leq v_j - v_i + M \cdot (1 - y_{ij}) \quad i, j \in N : i \neq j \quad (3.3k)$$

$$\hat{c}_{ij} \leq M \cdot y_{ij} \quad i, j \in N : i \neq j \quad (3.3l)$$

$$f_{ij} \leq x_{ij} \quad (i, j) \in E \quad (3.3m)$$

$$o_{ikj} \leq y_{ij} \quad (i, k, j) \in O : j \neq 1 \quad (3.3n)$$

$$\sum_{(i,k,j) \in O} o_{ikj} = 1 - \gamma_k \quad k \in N \quad (3.3o)$$

$$\sum_{r \in \delta_O^+(i)} o_r + \sum_{e \in \delta^+(i)} f_e = \sum_{r \in \delta_O^-(i)} o_r + \sum_{e \in \delta^+(i)} f_e \quad i \in N \quad (3.3p)$$

$$\sum_{r \in \delta_O^+(1)} o_r + \sum_{e \in \delta^+(1)} f_e = 1 \quad (3.3q)$$

$$w_{ikj} \geq (d_{ik} + d_{kj})o_{ikj} - \hat{c}_{ij} \quad (i, k, j) \in O : (i, j) \neq (1, 1) \quad (3.3r)$$

$$w_{1k1} \geq (d_{1k} + d_{k1})o_{1k1} - \sum_{e \in E} c_e x_e \quad k \in N \setminus \{1\} \quad (3.3s)$$

$$w \in \mathbb{R}_+^{|O|} \quad (3.3t)$$

$$f \in \{0, 1\}^{|E|} \quad (3.3u)$$

$$o \in \{0, 1\}^{|O|} \quad (3.3v)$$

$$\gamma \in \{0, 1\}^{|N|} \quad (3.3w)$$

$$y_{ij} \in \{0, 1\} \quad i, j \in N : i \neq j \quad (3.3x)$$

$$u, v \in \mathbb{R}_+^{|N|} \quad (3.3y)$$

$$\hat{c}_{ij} \geq 0. \quad i, j \in N : i \neq j \quad (3.3z)$$

(3.3a) reconstruye el vector  $(x, \gamma)$  y (3.3b)-(3.3d) son restricciones para registrar la posición del nodo  $i$  en la ruta del camión, propuestas por Miller et al. (1960). La variable auxiliar  $u_i \in \mathbb{Z}_+$  toma como valor la posición del nodo  $i \in N$  en la ruta del camión.

(3.3e)-(3.3g) construyen la variable  $y_{ij} \in \{0, 1\}$ , que toma valor 1 si y solo si el nodo  $i$  se visita antes que el nodo  $j$  en la ruta del camión, para  $i, j \in N$  con  $i \neq j$ .

(3.3h)-(3.3j) construyen la variable  $v_i \in \mathbb{R}$ , que toma como valor la duración acumulada de la ruta del camión desde el *depot* hasta el nodo  $i \in N$ . (3.3k)-(3.3l) construyen la variable  $\hat{c}_{ij} \in \mathbb{R}$ , cuyo valor es, a lo más, la duración acumulada de la ruta del camión entre los nodos  $i$  y  $j$ , para  $i, j \in N$  con  $i \neq j$ .  $\hat{c}_{ij}$  será 0 para cualquier par  $(i, j)$  tal que  $y_{ij} = 0$ . El valor de  $M$  utilizado es

$$M \triangleq \min_{(x, \gamma) \in \mathbb{X}} \left\{ \sum_{e \in E} c_e x_e : \gamma = \mathbf{1} \right\}.$$

La familia (3.3m) fuerza a que el *drone* solo pueda viajar a bordo del camión en los arcos de la ruta del camión. La familia (3.3n) garantiza que las operaciones comienzan en un nodo  $i$  y terminan en un nodo  $j$ , tal que  $i$  es visitado por el camión antes que  $j$ . La familia (3.3o) fuerza al *drone* a visitar todo nodo no visitado por el camión. La familia (3.3p) exige que a cada nodo se entre y se salga la misma cantidad de veces. (3.3q) fuerza a que el *drone* visite el *depot*. (3.3r)-(3.3t) construyen  $w_{ikj}$  como la parte positiva de la diferencia entre la duración del despacho directo a  $k$  del *drone* y la duración de la sección de la ruta del camión entre  $i$  y  $j$ . Finalmente, (3.3u)-(3.3z) definen la naturaleza de las variables.

La expresión (3.4) define la función objetivo del problema, para una ruta del camión  $(x, \gamma) \in \mathbb{X}$  y un vector  $(f, o, w) \in \mathbb{Y}(x)$ . Así, el objetivo corresponde a minimizar la duración total de la ruta del camión, esto es, la duración de la ruta del camión más la espera del mismo generada por la ruta del *drone*.

$$\sum_{e \in E} c_e x_e + \sum_{(i, k, j) \in \mathcal{O}} w_{ikj}. \quad (3.4)$$

#### 4. PROPIEDADES DEL PROBLEMA

En este capítulo se proponen propiedades que permiten fortalecer la formulación anterior.

PROPIEDAD 1. *Existe una solución óptima al TSP-D en la cual el drone no viaja dos arcos consecutivos a bordo del camión.*

DEMOSTRACIÓN. Considere una sección de tres nodos de la ruta del camión  $\{i, k, j\}$ . Se sabe que  $c$  define una métrica sobre  $N^2$ , por lo que  $c_{ij} \leq c_{ik} + c_{kj}$ . Además,  $\alpha \geq 1$ , por lo que  $d_{ik} + d_{kj} \leq c_{ik} + c_{kj}$ . Combinando estas desigualdades, obtenemos

$$\max\{c_{ij}, d_{ik} + d_{kj}\} \leq c_{ik} + c_{kj} . \quad (4.1)$$

De la ecuación (4.1) se desprende que la duración de la operación  $(i, k, j)$  es, a lo más, la duración de la ruta del camión  $\{i, k, j\}$  con el *drone* a bordo.  $\square$

COROLARIO 1. *Para  $n \geq 3$ , la solución óptima al TSP es subóptima con respecto a la solución óptima al TSP-D, sobre el mismo grafo.*

Como resultado del Corolario 1, y sin pérdida de optimalidad, se puede fortalecer la formulación de  $\mathbb{X}$  agregando la restricción

$$\sum_{i \in N} \gamma_i \leq n - 1$$

al sistema (3.2).

PROPIEDAD 2. *Existe una solución óptima al TSP-D en la cual todas las operaciones son consecutivas.*

DEMOSTRACIÓN. Considere una solución factible en que el *drone* viaja a bordo del camión entre dos operaciones. Sean  $i$  y  $j$  los nodos de aterrizaje de la primera operación

y de despegue de la segunda, respectivamente. En vista de la Propiedad 1, sin pérdida de optimalidad, se puede suponer que no hay nodos entre  $i$  y  $j$ , como detalla la Figura 4.1. La espera asociada a la primera operación ( $w_1$ ) está definida por el tiempo de vuelo del *drone* hacia el cliente ( $d_1$ ), el tiempo de vuelo del *drone* desde el cliente ( $d_2$ ) y la duración de la ruta parcial del camión entre el despegue y el aterrizaje ( $\hat{c}$ ), es decir,  $w_1 = (d_1 + d_2 - \hat{c})_+$ .

Sea  $d_3$  el tiempo de vuelo del *drone* desde el cliente atendido en la primera operación hacia  $j$ . Se cumple que

$$\begin{aligned}
w_1 &= (d_1 + d_2 - \hat{c})_+ \\
&\geq (d_1 + d_2 + d_{ij} - (\hat{c} + c_{ij}))_+ \\
&\geq (d_1 + d_3 - (\hat{c} + c_{ij}))_+ \\
&= w_2 ,
\end{aligned}$$

donde  $w_2$  es la espera asociada a una primera operación que posterga el retorno del *drone* hasta el nodo  $j$ . □

**COROLARIO 2.** *Existe una solución óptima al TSP-D en que el depot es nodo de despegue y aterrizaje para operaciones.*

**COROLARIO 3.** *Sin pérdida de optimalidad, se puede reemplazar (3.3m) de la formulación de  $\mathbb{Y}(x)$  por  $f = \mathbf{0}$ .*

El Corolario 3 permite eliminar la variable  $f$  y la restricción (3.3m) de la formulación (3.3).

Para las siguientes propiedades, considere la secuencia de nodos  $\mathcal{R}_x \triangleq \{r_0, r_1, \dots, r_{|\mathcal{R}_x|-1}\}$  ordenados según la ruta del camión  $(x, \gamma) \in \mathbb{X}$ . Además, considere una ruta de *drone* definida por los arcos en  $\mathcal{D}_x \subset E$  que minimiza la espera del camión que recorre  $\mathcal{R}_x$ . Se define, además, el conjunto  $\mathcal{O}_x \triangleq \{(i, k, j) \in \mathcal{O} : (i, k), (k, j) \in \mathcal{D}_x\}$  de operaciones formadas por los arcos en  $\mathcal{D}_x$ .

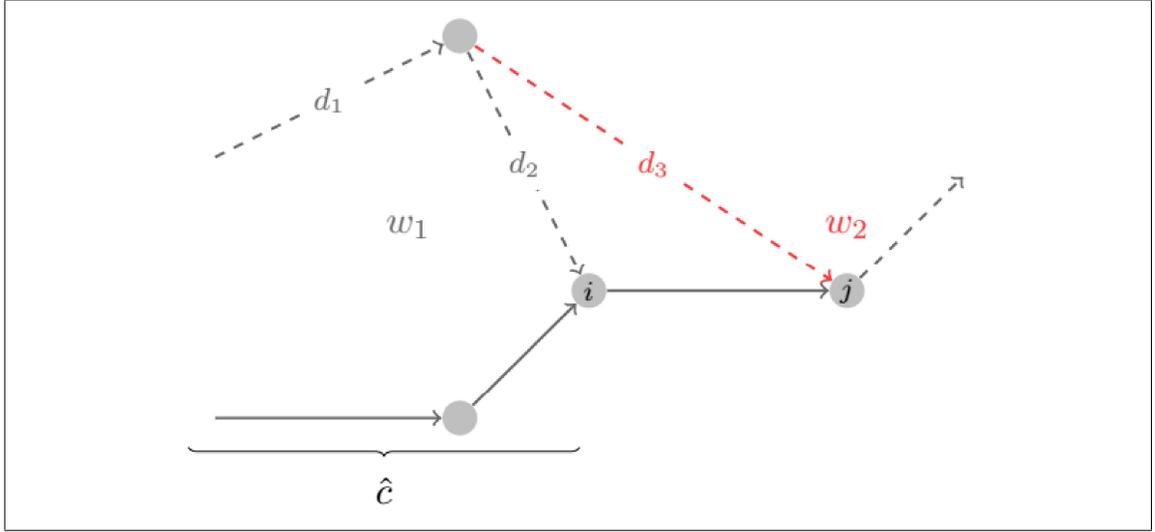


Figura 4.1. Ejemplo ilustrativo de la Propiedad 2

PROPIEDAD 3. La duración total de una ruta del camión  $(x, \gamma) \in \mathbb{X}$  está acotada inferiormente por la máxima duración entre las rutas  $\mathcal{R}_x$  y  $\mathcal{D}_x$ , i.e.

$$\max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in \mathcal{D}_x} d_e \right\} \leq \sum_{e \in E} c_e x_e + \sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x). \quad (4.2)$$

DEMOSTRACIÓN. Si  $\mathcal{O}_x = \emptyset$ , entonces el lado izquierdo de (4.2) se reduce a  $\sum_{e \in E} c_e x_e$  y es igual al lado derecho. Si  $\mathcal{O}_x \neq \emptyset$ , se cumple que

$$\begin{aligned} \max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in \mathcal{D}_x} d_e \right\} &= \sum_{e \in E} c_e x_e + \left( \sum_{e \in \mathcal{D}_x} d_e - \sum_{e \in E} c_e x_e \right)_+ \\ &= \sum_{e \in E} c_e x_e + \left[ \sum_{(i,k,j) \in \mathcal{O}_x} (d_{ik} + d_{kj} - \hat{c}_{ij}(\mathcal{R}_x)) \right]_+ \\ &\leq \sum_{e \in E} c_e x_e + \sum_{(i,k,j) \in \mathcal{O}_x} (d_{ik} + d_{kj} - \hat{c}_{ij}(\mathcal{R}_x))_+ \\ &= \sum_{e \in E} c_e x_e + \sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x), \end{aligned}$$

donde la segunda igualdad es consecuencia de la Propiedad 2 y la desigualdad proviene de la subaditividad de la función  $(\cdot)_+$ . □

A partir de la Propiedad 3, se propone generar una cota inferior para la espera del camión. Dada una ruta  $(x, \gamma) \in \mathbb{X}$ , los arcos en  $\mathcal{D}_x$  definen un ciclo que incluye el *depot* e intercala nodos en  $\mathcal{R}_x$  y en  $N \setminus \mathcal{R}_x$ . Entonces, un ciclo que cumple con estas condiciones y es de mínima duración, es una aproximación cuya duración acota inferiormente a la duración de la ruta definida por  $\mathcal{D}_x$ .

Sea  $\mathbb{W}(\gamma)$  el conjunto de rutas de *drone* que cumplen las condiciones anteriormente mencionadas. Considere las variables binarias  $s_e \in \{0, 1\}$ , que es igual a 1 si y solo si el arco  $e \in E$  pertenece a la aproximación,  $a_e \in \{0, 1\}$ , que es igual a 1 si y solo si el arco  $e \in E$  pertenece a la aproximación y sale de un nodo en  $\mathcal{R}_x$ , y  $b_e \in \{0, 1\}$ , que es igual a 1 si y solo si el arco  $e \in E$  pertenece a la aproximación y entra a un nodo en  $\mathcal{R}_x$ . Luego,  $\mathbb{W}(\gamma)$  corresponde al conjunto de vectores  $s$  para los cuales existe  $(a, b, \gamma')$  tal que

$$s_e = a_e + b_e \quad e \in E \quad (4.3a)$$

$$\sum_{e \in \delta^-(i)} a_e = \sum_{e \in \delta^+(i)} b_e = 1 - \gamma_i \quad i \in N \quad (4.3b)$$

$$\sum_{e \in \delta^+(i)} a_e = \sum_{e \in \delta^-(i)} b_e \quad i \in N \quad (4.3c)$$

$$\sum_{e \in \delta^+(i)} a_e \leq \gamma_i \quad i \in N \quad (4.3d)$$

$$\sum_{e \in \delta^-(i)} b_e \leq \gamma_i \quad i \in N \quad (4.3e)$$

$$\sum_{e \in \delta^+(1)} a_e = 1 \quad (4.3f)$$

$$\gamma'_i = \sum_{e \in \delta^+(i)} s_e \quad i \in N \quad (4.3g)$$

$$\sum_{e \in \delta^+(S)} s_e \geq \max_{i \in S} \{\gamma'_i\} \quad \emptyset \neq S : 1 \notin S \subset N \quad (4.3h)$$

$$s_{1j} \leq \sum_{(i,1) \in \delta^-(1): i \leq j} s_{i1} \quad (1, j) \in \delta^+(1) \quad (4.3i)$$

$$s, a, b \in \{0, 1\}^{|E|} \quad (4.3j)$$

$$\gamma' \in \{0, 1\}^{|N|}. \quad (4.3k)$$

(4.3a) construye  $s_e$  como la suma de  $a_e$  y  $b_e$ . (4.3b)-(4.3c) fuerzan la conservación de flujo y garantiza la visita a todos los nodos que están fuera de la ruta del camión. (4.3d)-(4.3e) fuerza a que los arcos seleccionados formen operaciones. (4.3f) fuerza a que el *depot* sea el nodo de partida de la ruta. (4.3g) define el vector  $\gamma'$ , el cual es análogo a  $\gamma$ , pero para la aproximación a la ruta del *drone*. (4.3i) elimina simetría, de la misma forma que lo hace (3.2c). Finalmente, (4.3h) elimina sub-ciclos y (4.3j)-(4.3k) definen la naturaleza de las variables.

La Figura 4.2 ejemplifica esta aproximación. En la Figura 4.2a se muestra la solución al TSP-D, donde la ruta del camión visita a los clientes  $C = \{2, 3, 4, 5, 6, 7, 12, 13, 15\}$ , dejando al *drone* la visita de los nodos  $N \setminus C$ . La Figura 4.2b muestra la ruta del *drone* que forma la aproximación. Esta última ruta no posee cruce de arcos, a diferencia de la ruta en la Figura 4.2a, pues sería suboptimal considerando que minimiza su duración.

La formulación (4.3) permite generar rutas de *drone* en las que las operaciones construidas no son necesariamente factibles con respecto a la ruta del camión, pues pueden no ser consistentes con su secuenciamiento. Si bien la aproximación mejora si se impone que esto no ocurra, (4.3) depende únicamente de  $\gamma$  y no de  $x$ . Es decir, dada una ruta del camión  $(x, \gamma) \in \mathbb{X}$ , una ruta del *drone*

$$\bar{s}(\gamma) \in \arg \min_{s \in \mathbb{W}(\gamma)} \left\{ \sum_{e \in E} d_e s_e \right\} \quad (4.4)$$

sirve como aproximación y, además, es común para toda ruta del camión  $(x, \gamma)$  con el mismo sub-vector  $\gamma$ . Esto se formaliza en la Propiedad 4.

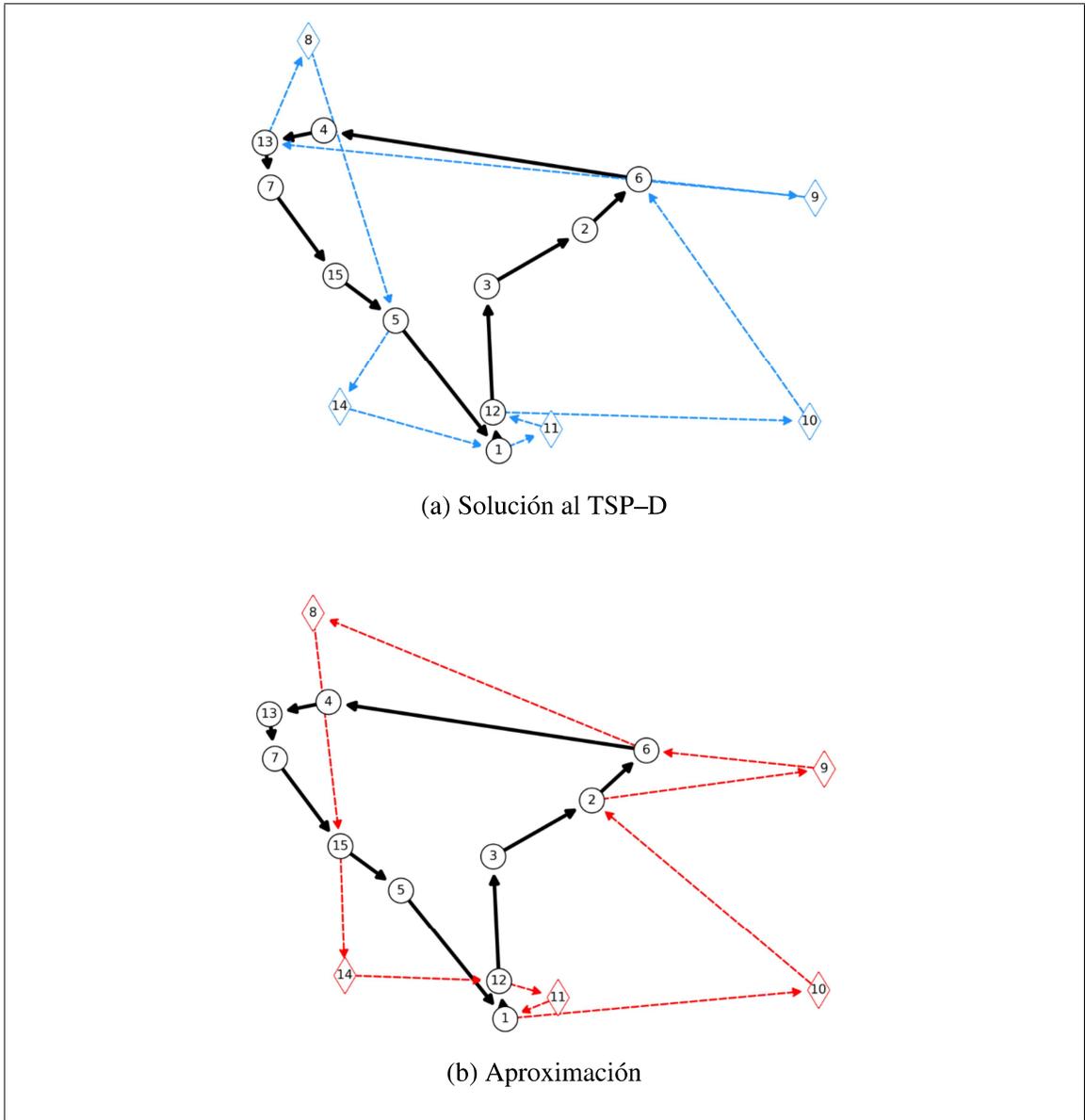


Figura 4.2. Ejemplo de la aproximación propuesta

PROPIEDAD 4. Para cualquier ruta del camión  $(x, \hat{\gamma}) \in \mathbb{X}$ , la desigualdad

$$\left( \sum_{e \in E} d_e \bar{s}_e(\hat{\gamma}) - c_e x_e \right)_+ \leq \sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x) \quad (4.5)$$

es válida sin pérdida de optimalidad, donde  $\bar{s}(\hat{\gamma})$  es una ruta del dron que cumple (4.4)

DEMOSTRACIÓN. Por definición, se tiene que para cualquier ruta  $(x, \hat{\gamma}) \in \mathbb{X}$  se cumple que

$$\sum_{e \in E} d_e \bar{s}_e(\hat{\gamma}) \leq \sum_{e \in \mathcal{D}_x} d_e . \quad (4.6)$$

Usando (4.2) y (4.6), se obtiene

$$\begin{aligned} \left( \sum_{e \in E} d_e \bar{s}_e(\hat{\gamma}) - c_e x_e \right)_+ &= \max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in E} d_e \bar{s}_e(\hat{\gamma}) \right\} - \sum_{e \in E} c_e x_e \\ &\leq \max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in \mathcal{D}_x} d_e \right\} - \sum_{e \in E} c_e x_e \\ &\leq \sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x) . \end{aligned}$$

□

Intuitivamente, a medida que una ruta del camión visite menos nodos, su espera será mayor, pues la ruta del *drone* visitará más nodos. Dada una ruta del camión, se puede garantizar esto, si una nueva ruta del camión que visita menos nodos conserva el secuenciamiento de la primera. Para formalizar esta idea, se definen los conjuntos *t*-atajos y *t*-reducción.

DEFINICION 1. *t*-atajos

Dada una ruta  $(\hat{x}, \hat{\gamma})$  y un entero  $t \in \{1, \dots, t_{\hat{\gamma}}^{\max}\}$ , donde  $t_{\hat{\gamma}}^{\max} \triangleq \sum_{i \in N} \hat{\gamma}_i - \lceil \frac{n}{2} \rceil$ , se define el conjunto de *t*-atajos o atajos de largo máximo *t*, denotado como  $\mathcal{S}_{\hat{x}}(t)$ , como el conjunto de arcos  $(i, j) \in E$  tal que *i* y *j* pertenecen a  $\mathcal{R}_x$ , *i* se visita antes que *j*, y se saltan, a lo más, *t* nodos consecutivos de la ruta del camión. Matemáticamente, se define como

$$\mathcal{S}_{\hat{x}}(t) \triangleq \{(r_i, r_j) \in \mathcal{R}_{\hat{x}}^2 : (r_i, r_j) \in E, 2 \leq j - i \leq 1 + t\} . \quad (4.7)$$

La Figura 4.3 ejemplifica, con arcos discontinuos, los elementos pertenecientes al conjunto de 1-atajos de una ruta del camión representada con arcos continuos. Si se consideran los arcos discontinuos y los arcos punteados, se obtiene el conjunto de 2-atajos.

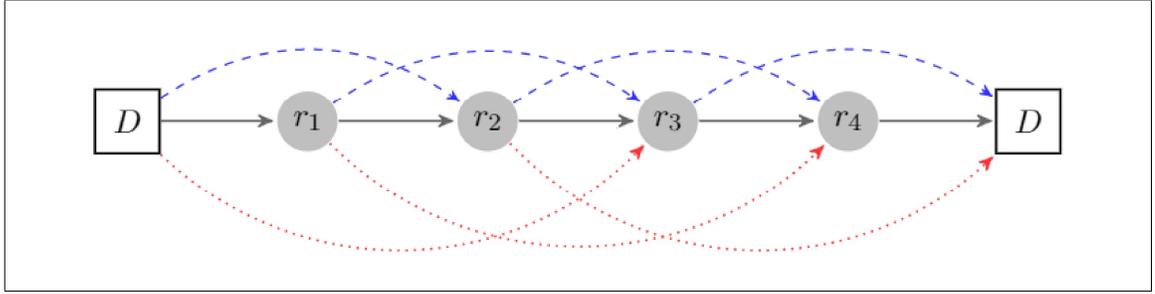


Figura 4.3. Ejemplo de arcos pertenecientes a 1-atajos y 2-atajos

PROPOSICIÓN 1. Dada una ruta del camión  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$ , toda ruta  $(x, \gamma) \in \mathbb{X}$  cumple

$$\sum_{e \in E: \hat{x}_e=1} (1 - x_e) - \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i=1} (1 - \gamma_i) \geq 0. \quad (4.8)$$

DEMOSTRACIÓN. (4.8) es equivalente a

$$\sum_{e \in E: \hat{x}_e=1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e \leq \sum_{i \in N: \hat{\gamma}_i=1} \gamma_i. \quad (4.9)$$

El lado izquierdo de (4.9) corresponde al número de arcos en la ruta  $(\hat{x}, \hat{\gamma})$  reutilizados en  $(x, \gamma)$  más el número de arcos en  $\mathcal{S}_{\hat{x}}(t)$  utilizados en  $(x, \gamma)$ , mientras que el lado derecho corresponde al número de nodos en  $\mathcal{R}_{\hat{x}}$  repetidos en  $\mathcal{R}_x$ .

Sean  $U \subseteq N$  y  $\Omega(U) \triangleq \{(i, j) \in E : i, j \in U\}$ . Dado que  $x$  define un ciclo sobre  $\mathcal{R}_x \subseteq N$ , se cumple que  $\sum_{e \in \Omega(U)} x_e \leq |U|$  para todo  $U \subseteq N$ . En particular, cuando  $U = \mathcal{R}_x \cap \mathcal{R}_{\hat{x}}$ , se tiene que

$$\sum_{e \in \Omega(\mathcal{R}_x \cap \mathcal{R}_{\hat{x}})} x_e \leq |\mathcal{R}_x \cap \mathcal{R}_{\hat{x}}|. \quad (4.10)$$

Por otro lado, se cumple que

$$\begin{aligned} \Omega(\mathcal{R}_x \cap \mathcal{R}_{\hat{x}}) &= \{(i, j) \in E : \hat{x}_{ij} = 1, i, j \in \mathcal{R}_x\} \\ &\cup \{(i, j) \in \mathcal{S}_{\hat{x}}(t) : i, j \in \mathcal{R}_x\} \\ &\cup \{(i, j) \in E \setminus \mathcal{S}_{\hat{x}}(t) : \hat{x}_{ij} = 0, i, j \in \mathcal{R}_x \cap \mathcal{R}_{\hat{x}}\}, \end{aligned}$$

por lo que se obtiene

$$\sum_{e \in E: \hat{x}_e=1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e \leq \sum_{e \in \Omega(\mathcal{R}_x \cap \mathcal{R}_{\hat{x}})} x_e. \quad (4.11)$$

Combinando (4.10) y (4.11), se obtiene que todo  $(x, \gamma) \in \mathbb{X}$  cumple

$$\sum_{e \in E: \hat{x}_e=1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e \leq |\mathcal{R}_x \cap \mathcal{R}_{\hat{x}}| = \sum_{i \in N: \hat{\gamma}_i=1} \gamma_i.$$

□

#### DEFINICION 2. $t$ -reducción

Dada una ruta del camión  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$  y un entero  $t \in \{1, \dots, t_{\hat{\gamma}}^{\max}\}$ , se define el conjunto de  $t$ -reducción, denotado como  $\mathcal{T}_{\hat{x}}(t)$ , como el conjunto de rutas que se forman al unir sub-secuencias de  $\mathcal{R}_{\hat{x}}$  manteniendo el secuenciamiento original. Es decir, una ruta  $(x, \gamma)$  pertenece a  $\mathcal{T}_{\hat{x}}(t)$  si y solo si es una ruta que cubre  $\mathcal{R}_x \subseteq \mathcal{R}_{\hat{x}}$  reutilizando arcos de  $(\hat{x}, \hat{\gamma})$  y, potencialmente, utilizando arcos en  $\mathcal{S}_{\hat{x}}(t)$ .

La Figura 4.4 ejemplifica cuatro rutas pertenecientes a una  $2$ -reducción de la ruta del camión presentada en la Figura 4.3. Para cada ruta  $(x, \gamma)$ ,  $\sum_{e \in E: \hat{x}_e=1} (1 - x_e)$  equivale a la cantidad de arcos continuos removidos,  $\sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e$  a la cantidad de arcos discontinuos o punteados, y  $\sum_{i \in N: \hat{\gamma}_i=1} (1 - \gamma_i)$  a la cantidad de nodos en  $\mathcal{R}_{\hat{x}}$  no visitados.

PROPOSICIÓN 2. Sean  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$  y  $t \in \{1, \dots, t_{\hat{\gamma}}^{\max}\}$ . Una ruta  $(x, \gamma) \in \mathcal{T}_{\hat{x}}(t)$  si y solo si cumple

$$\sum_{e \in E: \hat{x}_e=1} (1 - x_e) - \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i=1} (1 - \gamma_i) = 0, \quad (4.12)$$

donde  $\hat{\gamma}$  es tal que  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$ .

DEMOSTRACIÓN.

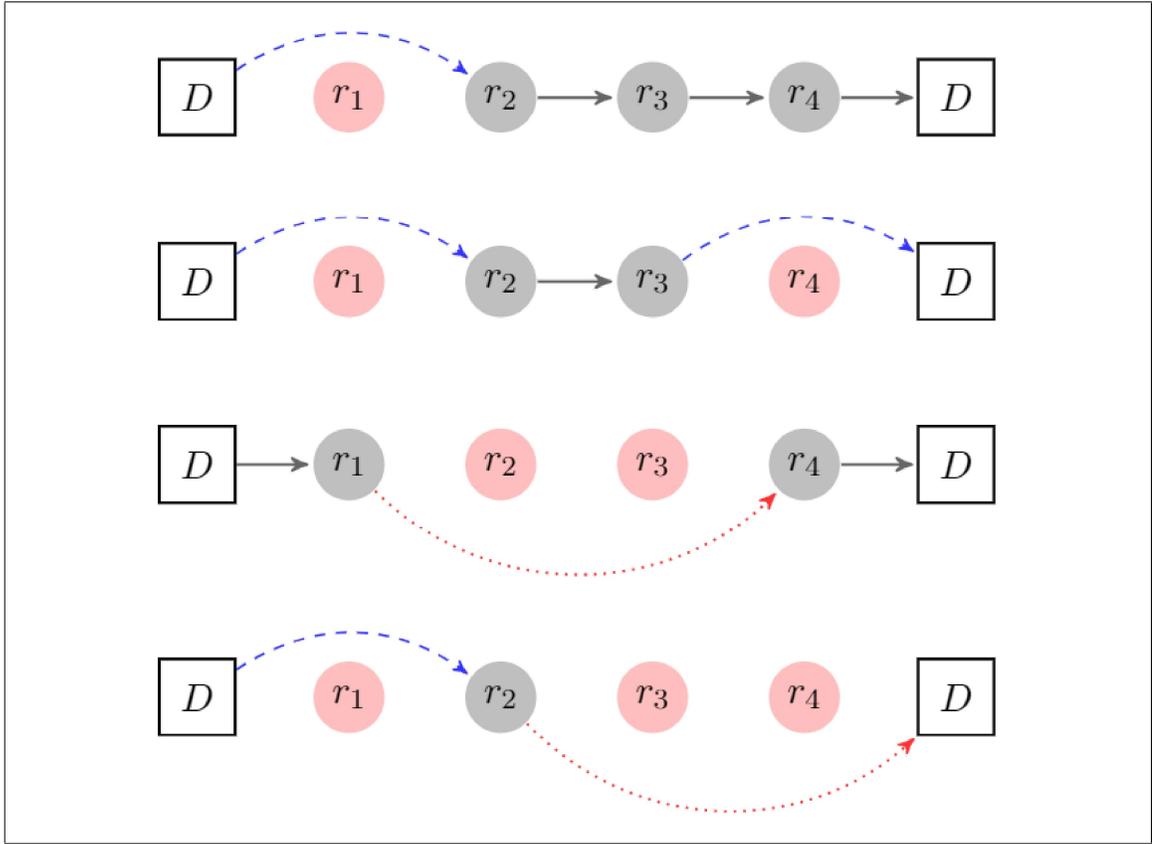


Figura 4.4. Ejemplo de rutas en una 2-reducción

Primero se prueba que si  $(x, \gamma) \in \mathcal{T}_{\hat{x}}(t)$ , entonces satisface (4.12). Sea  $U \subseteq \mathcal{R}_x$ . Se sabe que  $\sum_{e \in \Omega(U)} x_e = |U|$  si y solo si  $U = \mathcal{R}_x$ . Además, se tiene que  $\mathcal{S}_{\hat{x}}(t) \subseteq \{e \in E : \hat{x}_e = 0\}$ . Por definición, si  $(x, \gamma) \in \mathcal{T}_{\hat{x}}(t)$ , se cumple que

$$\sum_{e \in E \setminus \mathcal{S}_{\hat{x}}(t) : \hat{x}_e = 0} x_e = 0.$$

Luego, se obtiene que

$$\begin{aligned} \sum_{e \in E : \hat{x}_e = 1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e + \sum_{e \in \{r \in E : \hat{x}_r = 0\} \setminus \mathcal{S}_{\hat{x}}(t)} x_e & \overset{0}{=} \sum_{e \in E} x_e \\ & = \sum_{e \in \Omega(\mathcal{R}_x)} x_e \\ & = |\mathcal{R}_x| \end{aligned}$$

$$\begin{aligned}
&= |\mathcal{R}_x \cap \mathcal{R}_{\hat{x}}| \\
&= \sum_{i \in N: \hat{\gamma}_i = 1} \gamma_i,
\end{aligned}$$

por lo que  $(x, \gamma)$  cumple (4.12).

Por otro lado, se prueba que si  $(x, \gamma)$  satisface (4.12), entonces pertenece a  $\mathcal{T}_{\hat{x}}(t)$ . Equivalentemente, se prueba que si  $(x, \gamma) \in \mathbb{X} \setminus \mathcal{T}_{\hat{x}}(t)$ , entonces no satisface (4.12).

Si  $(x, \gamma) \in \mathbb{X} \setminus \mathcal{T}_{\hat{x}}(t)$ , entonces

$$\sum_{e \in E \setminus \mathcal{S}_{\hat{x}}(t): \hat{x}_e = 0} x_e \geq 1.$$

Aun más, se tiene que

$$\sum_{e \in \delta^+(\mathcal{R}_{\hat{x}} \cap \mathcal{R}_x)} x_e + \sum_{e \in \Omega(\mathcal{R}_{\hat{x}} \cap \mathcal{R}_x) \setminus \mathcal{S}_{\hat{x}}(t): \hat{x}_e = 0} x_e \geq 1.$$

Por ende, considerando la Proposición 1 y el hecho de que

$$\sum_{e \in E: \hat{x}_e = 1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e + \sum_{e \in \delta^+(\mathcal{R}_{\hat{x}} \cap \mathcal{R}_x)} x_e + \sum_{e \in \Omega(\mathcal{R}_{\hat{x}} \cap \mathcal{R}_x) \setminus \mathcal{S}_{\hat{x}}(t): \hat{x}_e = 0} x_e = \sum_{i \in N: \hat{\gamma}_i = 1} \sum_{e \in \delta^+(i)} x_e,$$

toda ruta  $(x, \gamma) \in \mathbb{X} \setminus \mathcal{T}_{\hat{x}}(t)$  cumple

$$\sum_{e \in E: \hat{x}_e = 1} x_e + \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e \leq \sum_{i \in N: \hat{\gamma}_i = 1} \gamma_i - 1. \quad (4.13)$$

Reordenando (4.13), se obtiene

$$\sum_{e \in E: \hat{x}_e = 1} (1 - x_e) - \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i = 1} (1 - \gamma_i) \geq 1.$$

Por lo tanto, toda ruta  $(x, \gamma)$  pertenece a  $\mathcal{T}_{\hat{x}}(t)$  si y solo si satisface (4.12).  $\square$

Como resultado de la Proposición 2, equivalentemente, se puede definir la  $t$ -reducción de  $(\hat{x}, \hat{\gamma})$  como

$$\mathcal{T}_{\hat{x}}(t) \triangleq \left\{ (x, \gamma) \in \mathbb{X} : \sum_{e \in E: \hat{x}_e=1} (1 - x_e) - \sum_{e \in \mathcal{S}_{\hat{x}}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i=1} (1 - \gamma_i) = 0 \right\}.$$

La mínima espera asociada a una ruta del camión  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$  es una cota inferior para la mínima espera de cualquier ruta perteneciente a su  $t$ -reducción. Esto se formaliza en la Propiedad 5.

PROPIEDAD 5. *Dada un ruta  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$  y un entero  $t \in \{1, \dots, t_{\hat{\gamma}}^{\max}\}$ , el corte*

$$\sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x) \geq \sum_{(i,k,j) \in \mathcal{O}_{\hat{x}}} w_{ikj}(\mathcal{R}_{\hat{x}}) \quad (4.14)$$

es válido para toda ruta  $(x, \gamma) \in \mathcal{T}_{\hat{x}}(t)$ .

DEMOSTRACIÓN. Si  $(\hat{x}, \hat{\gamma}) = (x, \gamma)$ , la demostración es trivial. Cualquier ruta  $(x, \gamma) \in \mathcal{T}_{\hat{x}}(t) \setminus \{(\hat{x}, \hat{\gamma})\}$  se forma removiendo, al menos, un nodo de  $\mathcal{R}_{\hat{x}}$ . Por lo tanto, se tiene que  $\hat{c}_{ij}(\mathcal{R}_x) \leq \hat{c}_{ij}(\mathcal{R}_{\hat{x}})$  para todo par  $i, j \in \mathcal{R}_{\hat{x}}$  y, luego,  $w_{ikj}(\mathcal{R}_x) \geq w_{ikj}(\mathcal{R}_{\hat{x}})$  para todo par  $i, j \in \mathcal{R}_{\hat{x}}$  y  $k \in N \setminus \mathcal{R}_{\hat{x}}$ . Además, las operaciones de una ruta de dron factible para  $(\hat{x}, \hat{\gamma})$  se puede obtener de aquellas en  $\mathcal{O}_x$  que cubren los nodos en  $\mathcal{R}_{\hat{x}}$ . Considerando esto, se tiene que

$$\begin{aligned} Q(x) &= \sum_{(i,k,j) \in \mathcal{O}_x} w_{ikj}(\mathcal{R}_x) \\ &= \sum_{(i,k,j) \in \mathcal{O}_x: k \in \mathcal{R}_{\hat{x}}} w_{ikj}(\mathcal{R}_x) + \sum_{(i,k,j) \in \mathcal{O}_x: k \in N \setminus \mathcal{R}_{\hat{x}}} w_{ikj}(\mathcal{R}_x) \\ &\geq \sum_{(i,k,j) \in \mathcal{O}_x: k \in N \setminus \mathcal{R}_{\hat{x}}} w_{ikj}(\mathcal{R}_x) \\ &\geq \sum_{(i,k,j) \in \mathcal{O}_x: k \in N \setminus \mathcal{R}_{\hat{x}}} w_{ikj}(\mathcal{R}_{\hat{x}}) \\ &\geq Q(\hat{x}). \end{aligned}$$

Así,  $(x, \gamma)$  satisfacen (4.14).

□

## 5. METODOLOGÍA

Las formulaciones del TSP–D presentes en la literatura, a través de resolución directa, reportan soluciones para instancias pequeñas, por lo que, intuitivamente, una estrategia de descomposición podría ser apropiada para resolver instancias de mayor tamaño. En este capítulo, se presenta una adaptación para el TSP–D de un método de solución basado en descomposición, como los propuestos por Benders (1962) y Laporte & Louveaux (1993).

Utilizando la notación definida en el Capítulo 3, el modelo completo del TSP–D está definido por (5.1) y será referido como el modelo “extendido”

$$\min \sum_{e \in E} c_e x_e + \sum_{(i,k,j) \in O} w_{ikj} \quad (5.1a)$$

$$\text{s. a. } (x, \gamma) \in \mathbb{X} \quad (5.1b)$$

$$(f, o, w) \in \mathbb{Y}(x) . \quad (5.1c)$$

La descomposición propuesta consiste en dividir (5.1) en un problema maestro (MP), en el cual se planifica una ruta del camión  $(x, \gamma)$ , y un subproblema dependiente de  $x$ , en el cual se define una ruta óptima del *drone* para esa ruta  $(x, \gamma)$  del camión. El problema maestro está definido en (5.2),

$$\text{(MP)} \quad \min \sum_{e \in E} c_e x_e + Q(x) \quad (5.2a)$$

$$\text{s. a. } (x, \gamma) \in \mathbb{X} , \quad (5.2b)$$

donde  $Q(x)$  es una función que retorna la mínima espera del camión, dada una ruta  $x$ , definida por el subproblema (5.3)

$$Q(x) = \min \sum_{(i,k,j) \in O} w_{ikj} \quad (5.3a)$$

$$\text{s. a. } (f, o, w) \in \mathbb{Y}(x) . \quad (5.3b)$$

El método de descomposición consiste en reformular (MP) introduciendo una variable  $\theta$  que sobreestima  $Q(x)$  para todo  $(x, \gamma) \in \mathbb{X}$ . Esta reformulación de (MP) es

$$(MP) \quad \min \sum_{e \in E} c_e x_e + \theta \quad (5.4a)$$

$$\text{s. a. } (x, \gamma) \in \mathbb{X} \quad (5.4b)$$

$$\theta \geq Q(x) \quad (5.4c)$$

$$\theta \geq 0. \quad (5.4d)$$

(5.4c) puede ser reemplazado por un conjunto de restricciones lineales. Sin embargo, dicho conjunto podría contener un gran número de restricciones (una por ruta del camión factible, en el peor caso) y esto impide una solución explícita de (5.4). Alternativamente, se puede generar dichas restricciones dinámicamente cada vez que se encuentre una solución  $(\hat{x}, \hat{\gamma}, \hat{\theta})$  que las viole.

Sea (RMP) una relajación de (MP) relajando (5.4c) y definida por (5.5)

$$(RMP) \quad \min \sum_{e \in E} c_e x_e + \theta \quad (5.5a)$$

$$\text{s. a. } (x, \gamma) \in \mathbb{X} \quad (5.5b)$$

$$\theta \geq 0. \quad (5.5c)$$

La Propiedad 4 y la definición de  $\mathbb{W}(\gamma)$  en (4.3) implican que, para todo  $(x, \gamma) \in \mathbb{X}$ , existe  $s \in \mathbb{W}(\gamma)$  tal que

$$\left( \sum_{e \in E} d_e s_e - c_e x_e \right)_+ \leq Q(x) \leq \theta. \quad (5.6)$$

Así, se puede fortalecer (RMP) agregando (5.6) y obteniendo finalmente el modelo (5.7)

$$(RMP) \quad \min \quad \sum_{e \in E} c_e x_e + \theta \quad (5.7a)$$

$$\text{s. a.} \quad (x, \gamma) \in \mathbb{X} \quad (5.7b)$$

$$s \in \mathbb{W}(\gamma) \quad (5.7c)$$

$$\theta \geq \sum_{e \in E} d_e s_e - c_e x_e \quad (5.7d)$$

$$\theta \geq 0. \quad (5.7e)$$

Adicionalmente, se define a (LRMP) como la relajación lineal continua de (RMP). A continuación, se presentan desigualdades válidas que se generan y agregan dinámicamente a (LRMP) en el árbol de B&B, para reducir los tiempos de cómputo.

### 5.1. Cortes de optimalidad

Considere una solución  $(\hat{x}, \hat{\gamma})$  factible, y potencialmente óptima, a (RMP). Una familia de cortes que corrige la subestimación de  $\theta$  es la propuesta por Laporte & Louveaux (1993), definida por (5.8),

$$\theta \geq -Q(\hat{x})\Delta(x, \hat{x}) + Q(\hat{x}), \quad (5.8)$$

donde

$$\Delta(x, \hat{x}) \triangleq \sum_{e \in E: \hat{x}_e = 1} (1 - x_e) + \sum_{e \in E: \hat{x}_e = 0} x_e$$

es la *distancia de Hamming* (Robinson, 2008) entre  $x$  y  $\hat{x}$ .

Cuando  $x = \hat{x}$ , se cumple que  $\Delta(x, \hat{x}) = 0$  y (5.8) colapsa a  $\theta \geq Q(\hat{x})$ . En otro caso,  $\Delta(x, \hat{x}) = m \geq 1$ , por lo que (5.8) es equivalente a  $\theta \geq 0 \geq (1 - m)Q(\hat{x})$ . En una ruta, cada solución diferente a  $\hat{x}$  debe remover, al menos, un arco de  $\{e \in E : \hat{x}_e = 1\}$ , pues  $\hat{x}$

define un ciclo. Así, se puede reemplazar (5.8) por

$$\theta \geq -Q(\hat{x}) \sum_{e \in E: \hat{x}_e=1} (1 - x_e) + Q(\hat{x}), \quad (5.9)$$

que reduce considerablemente el número de variables involucradas en el corte.

Adicionalmente, utilizando la Propiedad 5, se puede refinar (5.9) aumentando su lado derecho para las soluciones en la  $t$ -reducción de  $(\hat{x}, \hat{\gamma})$ . El corte propuesto es de la forma

$$\theta \geq -Q(\hat{x}) \left[ \sum_{e \in E: \hat{x}_e=1} (1 - x_e) - \sum_{e \in S_{(\hat{x}, \hat{\gamma})}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i} (1 - \gamma_i) \right] + Q(\hat{x}), \quad (5.10)$$

donde  $t$  es un parámetro a configurar en  $\{1, \dots, t_{\hat{\gamma}}^{max}\}$ . El corte establece que, para cualquier solución factible  $(x, \gamma, \theta)$  a (RMP) tal que  $(x, \gamma)$  es elemento de la  $t$ -reducción de  $(\hat{x}, \hat{\gamma}, \hat{\theta})$ , se tiene que  $\theta \geq Q(\hat{x})$ . Esto, pues cualquier solución  $(x, \gamma)$  en la  $t$ -reducción de  $(\hat{x}, \hat{\gamma})$  es tal que el camión cubre un subconjunto de los nodos cubiertos por  $\hat{x}$  y conserva su secuenciamiento. Por lo mismo, el *drone* posee una carga adicional y la espera que genera para el camión debe ser, al menos,  $Q(\hat{x})$ .

## 5.2. Cortes correctivos

Como resultado de la Propiedad 4, se define una función correctiva  $g : \{0, 1\}^n \rightarrow \{0, 1\}^{|E|}$ , la cual retorna una ruta óptima de *drone* de la aproximación (4.4), dado  $(\hat{x}, \hat{\gamma}, \hat{\theta})$ .

Así, se genera el corte correctivo de la forma

$$(1 - s_e) \leq \Delta(\gamma, \hat{\gamma}) \quad e \in E : g_e(\hat{\gamma}) = 1. \quad (5.11)$$

que fuerza a que las rutas aproximadas del *drone*, para rutas del camión que cubren el mismo conjunto de nodos, sean iguales. Por ende, se aplica cada vez que se obtiene un valor de  $\gamma$  diferente al resolver (RMP). Para regular la cantidad de cortes añadidos a (RMP),

se puede agregar (5.11) y se reemplaza por

$$\sum_{e \in E: g_e(\hat{\gamma})=1} (1 - s_e) \leq \Delta(\gamma, \hat{\gamma}) \sum_{e \in E} g_e(\hat{\gamma}). \quad (5.12)$$

La elección entre (5.11) y (5.12) depende del balance existente entre las mejoras de cómputo que produce el corte más refinado contra la cantidad de cortes agregados, y su impacto negativo en la matriz de restricciones.

### 5.3. Eliminación de subtours

La familia de restricciones que eliminan sub-ciclos de camión que no pasan por el *depot*, es decir, (3.2b), se debe generar dinámicamente a medida que sus restricciones son requeridas. Para ello, se resuelven dos problemas de separación, dado un vector  $(\hat{x}, \hat{\gamma})$ , uno para soluciones enteras y otro para fraccionarias. En el primer caso, utilizando el algoritmo de Tarjan (Tarjan, 1972), se buscan componentes fuertemente conexas que estén desconectadas del *depot*. En el segundo caso, utilizando un algoritmo *Pre-flow push* (Ahuja et al., 1993), se buscan todas aquellas componentes que violen (3.2b) y se agrega dicha restricción para la componente que más la viole. Esto es, la componente  $S^*$  que cumple

$$S^* \in \arg \min_{\emptyset \neq S \subset N: \{1\} \notin S} \left\{ \sum_{e \in \delta^+(S)} \hat{x}_e - \max_{i \in S} \{\hat{\gamma}_i\} \right\}.$$

Para soluciones fraccionarias, este corte se agrega únicamente en el nodo raíz del árbol de B&B.

### 5.4. Estructura del método

La metodología propuesta funciona de la siguiente forma. Se resuelve (RMP) en un esquema similar a B&B, es decir, generando un árbol de búsqueda de problemas lineales continuos o nodos del árbol. Para ello, se ramifica (*branching*) un nodo del árbol cuya solución es fraccionaria y se poda (*pruning*) todo nodo cuya solución es entera, no posee

solución factible, o que posee una duración mayor al incumbente. En cada iteración, se revisa una lista  $T$  de nodos del árbol de B&B que aún no han sido resueltos. Cada vez que se ramifica, se añaden dos nodos nuevos a  $T$ . Y cada vez que se resuelve un nodo del árbol, se elimina de  $T$ .

Para garantizar que una solución  $(\hat{x}, \hat{\gamma}, \hat{\theta})$  sea factible para (MP), se agregan cortes de optimalidad cuando  $\hat{\theta} < Q(\hat{x})$ , garantizando que la solución obtenida sea descartada. Para acelerar su búsqueda, se insertan cortes correctivos en todo nodo en  $T$ . El método finaliza cuando ya no hay nodos en el árbol de búsqueda, *i.e.*,  $T = \emptyset$ , retornando la solución óptima a (MP). Un resumen del algoritmo de solución está descrito en pseudocódigo en el Algoritmo 1.

---

**Algoritmo 1:** Método de descomposición para resolver el TSP–D

---

**Input:**  $N, E, c, d$ , solución inicial  $(\bar{x}, \bar{\gamma}, \bar{\theta}) : \bar{\theta} \gg 0$ ,  
 $V = \emptyset$ , lista de nodos del árbol  $T = \{1\}$ .

**Output:** Solución óptima  $(x^*, \gamma^*)$  a (MP)

```

1 while  $T \neq \emptyset$  do
2   | Seleccionar nodo de  $T$  y resolver su (LRMP). Obtener solución  $(\hat{x}, \hat{\gamma}, \hat{\theta})$ ;
3   | if infeasible then
4   |   | Podar nodo;
5   | else if  $c^\top \hat{x} + \hat{\theta} \geq c^\top \bar{x} + \bar{\theta}$  then
6   |   | Podar nodo;
7   | else if  $(\hat{x}, \hat{\gamma})$  fraccionario then
8   |   | Ramificar e insertar dos nodos nuevos a  $T$ ;
9   |   | Remover nodo de  $T$ ;
10  | else
11  |   | CorteCorrectivo();
12  |   | Factible  $\leftarrow$  CorteOpt();
13  |   | if Factible then
14  |   |   | Actualizar incumbente:  $(\bar{x}, \bar{\gamma}, \bar{\theta}) \leftarrow (\hat{x}, \hat{\gamma}, \hat{\theta})$ ;
15  |   |   | Podar nodo;
16 return  $(\hat{x}, \hat{\gamma})$ ;

```

---

En las líneas 11 y 12 se invoca a las funciones generadoras de cortes. Es posible que estas sean invocadas por distintas soluciones a (LRMP) que comparten el mismo subvector

$(x, \gamma)$  o  $\gamma$ . Por ello, se registra una lista  $V$  de soluciones enteras a (RMP) para evitar re-calcular  $Q(x)$  o  $g(\gamma)$ .

El Algoritmo 2 detalla en pseudocódigo la función de cortes correctivos, mientras que el Algoritmo 3 la función de cortes de optimalidad.

---

**Algoritmo 2:** Función de corte correctivo

---

**Input:** Solución candidata entera  $(\hat{x}, \hat{\gamma}, \hat{\theta})$ , lista de soluciones enteras conocidas  $V$ .

1 **if**  $\nexists (x, \gamma) \in V : \gamma = \hat{\gamma}$  **then**  
 2     |    Calcular  $g(\hat{\gamma})$ ;  
 3     |    Añadir corte correctivo (5.12) a (LRMP);

---



---

**Algoritmo 3:** Función de corte de optimalidad

---

**Input:** Solución candidata entera  $(\hat{x}, \hat{\gamma}, \hat{\theta})$ , lista de soluciones enteras conocidas  $V$ .

**Output:** **Factible** si  $((\hat{x}, \hat{\gamma}, \hat{\theta}))$  es factible en (MP), **Infactible** en otro caso.

1 **if**  $(\hat{x}, \hat{\gamma}) \in V$  **then**  
 2     |    **return Factible**;  
 3 **else**  
 4     |     $V \leftarrow V \cup \{(\hat{x}, \hat{\gamma})\}$ ;  
 5     |    Calcular  $Q(\hat{x})$ ;  
 6     |    **if**  $\hat{\theta} < Q(\hat{x})$  **then**  
 7     |     |    Añadir corte de optimalidad (5.10) a (LRMP);  
 8     |     |    **return Infactible**;  
 9     |    **else**  
 10    |    |    **return Factible**;

---

## 6. EXTENSIONES DEL PROBLEMA

En este capítulo se presentan tres problemas que extienden nuestra formulación al TSP-D: TSP-D que admite *loops*, TSP-D con restricción de autonomía de vuelo del *drone* y TSP-D con múltiples *drones*.

### 6.1. TSP-D que admite *loops*

Este problema surge al levantar el supuesto (iii) que fuerza a que el *drone* aterrice y despegue en diferentes nodos del grafo. Así, el camión puede esperar en un mismo nodo mientras despliega una operación de *drone* que va y vuelve al mismo nodo, *i.e.*,  $(i, k, i)$ . Dichas operaciones son llamadas *loops*. Para abordar este problema, se adiciona una copia del *depot* y se indexa como  $n + 1$ , redefiniendo  $N = \{1, \dots, n + 1\}$  y el conjunto de arcos  $E$  de forma acorde. Esto se realiza para diferenciar las soluciones con *loops* que comienzan en el *depot* de las soluciones con solo una operación que comienza y termina en el *depot*.

Se extiende  $O$  de forma tal que  $(i, k, i) \in O$ ,  $i \in N$ , y se reformulan los conjuntos  $\mathbb{X}$ ,  $\mathbb{Y}(x)$  y  $\mathbb{W}(\gamma)$ . En la formulación de  $\mathbb{X}$  se relaja la restricción (3.2e). En la formulación de  $\mathbb{Y}(x)$  se agregan las variables  $o_{iki} \in \{0, 1\}$  para  $i, k \in N : i \neq k$ , y se reemplazan las restricciones (3.3n), (3.3p)-(3.3s) por

$$o_{ikj} \leq y_{ij} \quad (i, k, j) \in O : i \neq j \quad (6.1a)$$

$$\sum_{(i,k,j) \in \delta_O^+(i): i \neq j} o_{ikj} + \sum_{e \in \delta^+(i)} f_e = \sum_{(i,k,j) \in \delta_O^+(i): i \neq j} o_{ikj} + \sum_{e \in \delta^+(i)} f_e \quad i \in N \setminus \{1, n + 1\} \quad (6.1b)$$

$$\sum_{e \in \delta_O^+(1)} o_e + \sum_{e \in \delta^+(1)} f_e \geq 1 \quad (6.1c)$$

$$\sum_{e \in \delta_O^+(i)} o_e \leq n \cdot \gamma_i \quad i \in N \quad (6.1d)$$

$$w_{ikj} \geq o_{ikj} \cdot (d_{ik} + d_{kj}) - \hat{c}_{ij} \quad (i, k, j) \in O : i \neq j \quad (6.1e)$$

$$w_{iki} \geq o_{iki} \cdot (d_{ik} + d_{ki}) \quad (i, k, i) \in O \quad (6.1f)$$

En la formulación de  $\mathbb{W}(\gamma)$  se agrega la variable  $\ell_{ik} \in \{0, 1\}$ , que toma valor 1 si y solo si se realiza un *loop*  $(i, k, i)$  en la aproximación a la ruta del *drone*. Luego, se reformula como

$$\sum_{e \in \delta^-(k)} a_e = \sum_{e \in \delta^+(k)} b_e = 1 - \gamma_k \quad k \in N \setminus \{1, n+1\} \quad (6.2a)$$

$$\sum_{e \in \delta^+(i)} a_e \leq (n-1) \cdot \gamma_i \quad i \in N \setminus \{n+1\} \quad (6.2b)$$

$$\sum_{e \in \delta^-(i)} b_e \leq (n-1) \cdot \gamma_i \quad i \in N \setminus \{n+1\} \quad (6.2c)$$

$$\sum_{e \in \delta^+(1)} a_e \geq 1 \quad (6.2d)$$

$$\sum_{e \in \delta^-(n+1)} b_e = 1 \quad (6.2e)$$

$$\sum_{e \in \delta^+(n+1)} a_e = 0 \quad (6.2f)$$

$$\ell_{ik} = a_{ik} b_{ki} \quad i, k \in N : i \neq k \quad (6.2g)$$

$$\ell_e \leq \sum_{r \in \delta^+(i)} (a_r - \ell_r) \quad i \in N \setminus \{n+1\}, e \in \delta^+(i) \quad (6.2h)$$

$$s_{ij} = a_{ij} + b_{ij} - \ell_{ij} - \ell_{ji} \quad e \in E \quad (6.2i)$$

$$\gamma'_i = \sum_{e \in \delta^+(i)} s_e \quad i \in N \setminus \{1, n+1\} \quad (6.2j)$$

$$\sum_{e \in \delta^+(S)} s_e \geq \max_{i \in S} \{\gamma'_i\} \quad \emptyset \neq S \subset N : i, n+1 \notin S \quad (6.2k)$$

$$a, b, s, \ell \in \{0, 1\}^{|E|} \quad (6.2l)$$

$$\gamma'_i \in \{0, 1\} . \quad i \in N \setminus \{1, n+1\} \quad (6.2m)$$

(6.2a) cumple la misma función que (4.3b). (6.2b) y (6.2c) cumplen la misma función que (4.3d) y (4.3e), respectivamente, pero permitiendo la entrada y salida de más de un arco en cada nodo de la ruta del camión. (6.2d) exige que se salga, al menos, una vez del *depot*. (6.2e) y (6.2f) fuerzan a que la ruta termine en el *depot*  $n+1$ . (6.2g) define un *loop*. (6.2h) fuerza a que la solución contenga, al menos, una operación que no sea un *loop*. (6.2i) define  $s_{ij} \in \{0, 1\}$  como variable binaria que toma valor 1 si y solo si se realiza una operación de la forma  $(i, k, j)$ , con  $i \neq j$ . (6.2j) y (6.2k) cumplen la misma función que (4.3g) y (4.3h), respectivamente. Por último, (6.2l) y (6.2m) definen la naturaleza de las variables.

En este caso, la aproximación a la ruta del *drone* posee duración

$$\sum_{(i,j) \in E} [d_{ij}s_{ij} + \ell_{ij} \cdot (d_{ij} + d_{ji})] ,$$

por lo que se reemplaza (5.7d) por

$$\theta \geq \sum_{(i,j) \in E} [d_{ij}s_{ij} + \ell_{ij} \cdot (d_{ij} + d_{ji}) - c_{ij}x_{ij}] .$$

La Figura 6.1 ilustra la solución al TSP-D- $\ell$  para la instancia del ejemplo en la Figura 1.1.

## 6.2. TSP-D con restricción de autonomía de vuelo del *drone*

Otro problema a estudiar se obtiene al levantar el supuesto (vii) que asume autonomía de vuelo infinita para el *drone*. En este problema, denominado  $L$ -TSP-D, el *drone* posee una autonomía de  $L > 0$  unidades de tiempo por operación, de forma tal que cualquier

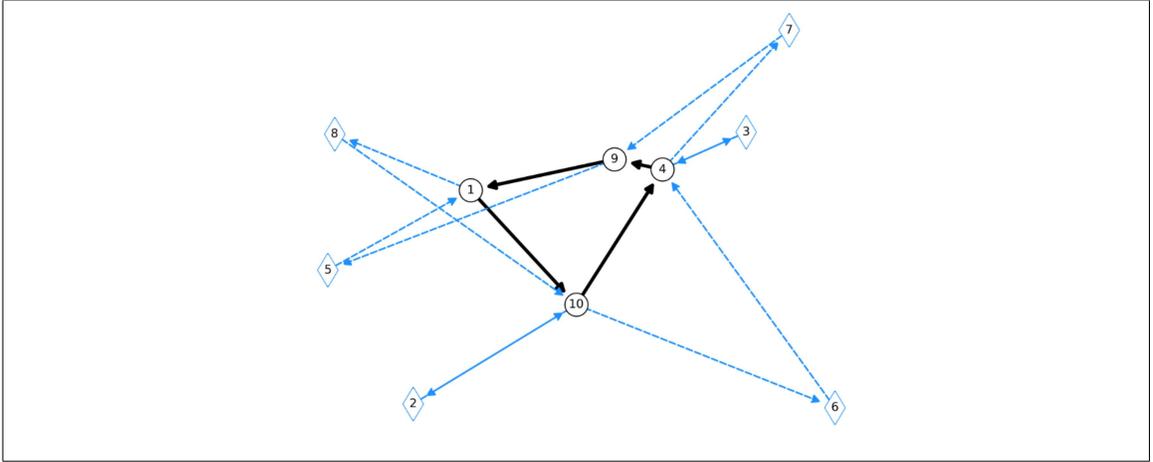


Figura 6.1. Solución óptima al TSP-D-ℓ

operación factible  $(i, k, j)$  debe cumplir que  $d_{ik} + d_{kj} \leq L$ . Por ello, se redefine

$$O = \{(i, k, j) \in N^3 : i \neq k, i \neq j, k \neq j, k \neq 1, d_{ik} + d_{kj} \leq L\} \\ \cup \{(1, k, 1) : k \in N \setminus \{1\}, d_{1k} + d_{k1} \leq L\}.$$

Dada una ruta del camión  $(\hat{x}, \hat{\gamma}) \in \mathbb{X}$ , es posible que no exista una ruta del *drone* factible, es decir, que (5.3) sea infactible. En esta caso, el Algoritmo 1 se corrige removiendo dicha solución y todo elemento en su *t-reducción* a través del corte de factibilidad

$$\sum_{e \in E: \hat{x}_e = 1} (1 - x_e) - \sum_{e \in \mathcal{S}_{(\hat{x}, \hat{\gamma})}(t)} x_e - \sum_{i \in N: \hat{\gamma}_i} (1 - \gamma_i) \geq 1. \quad (6.3)$$

Además, en este problema no se posee la Propiedad 2. Por ende, se debe considerar la familia de restricciones (3.3m) y la variable  $f$  en la formulación de  $\mathbb{Y}(x)$ . De la misma forma, en la formulación de  $\mathbb{W}(\gamma)$  se agrega la variable binaria  $f'_e \in \{0, 1\}$ , que toman valor 1 si y solo si el *drone* viaja a bordo del camión a través del arco  $e \in E$  en la aproximación a la ruta del *drone*, y se reformula como

$$s_e = a_e + b_e + f'_e \quad e \in E \quad (6.4a)$$

$$\sum_{e \in \delta^-(i)} a_e = \sum_{e \in \delta^+(i)} b_e = 1 - \gamma_i \quad i \in N \quad (6.4b)$$

$$\sum_{e \in \delta^+(i)} a_e + f'_e = \sum_{e \in \delta^-(i)} b_e + f'_e \quad i \in N \quad (6.4c)$$

$$\sum_{e \in \delta^+(i)} a_e + f'_e \leq \gamma_i \quad i \in N \quad (6.4d)$$

$$\sum_{e \in \delta^-(i)} b_e + f'_e \leq \gamma_i \quad i \in N \quad (6.4e)$$

$$f'_{ij} \leq \min\{\gamma_i, \gamma_j\} \quad (i, j) \in E \quad (6.4f)$$

$$\sum_{e \in \delta^+(1)} a_e + f'_e = 1 \quad (6.4g)$$

$$\sum_{e \in \delta^-(i)} a_e d_e + \sum_{e \in \delta^+(i)} b_e d_e \leq L \quad i \in N \quad (6.4h)$$

$$\gamma'_i = \sum_{e \in \delta^-(i)} s_e \quad i \in N \quad (6.4i)$$

$$\sum_{e \in \delta^+(S)} s_e \geq \max_{i \in S} \{\gamma'_i\} \quad \emptyset \neq S \subset N : 1 \notin S \quad (6.4j)$$

$$a, b, f' \in \{0, 1\}^{|E|} \quad (6.4k)$$

$$\gamma' \in \{0, 1\}^{|N|} . \quad (6.4l)$$

(6.4a) construye la variable  $s_e \in \{0, 1\}$  binaria que toma valor 1 si y solo si el *drone* viaja en el arco  $e \in E$ . (6.4b) cumple la misma función que (4.3b), mientras que (6.4c) fuerza a que se conserve flujo en los nodos de la ruta del camión. (6.4d) y (6.4e) cumplen la misma función que (4.3d) y (4.3e), respectivamente. (6.4f) fuerza a que el *drone* viaje a bordo del camión solo entre nodos de la ruta del camión. (6.4g) cumple la misma función que (4.3f). (6.4h) restringe la elección de operaciones a aquellas cuya duración es, a lo más,  $L$ . (6.4i) y (6.4j) cumplen la misma función que (4.3g) y (4.3h), respectivamente. Por último, (6.4k) y (6.4l) definen la naturaleza de las variables.

En este caso, la aproximación a la ruta del *drone* posee duración

$$\sum_{e \in E} [d_e \cdot (a_e + b_e) + c_e \cdot f'_e] ,$$

por lo que se reemplaza (5.7d) por

$$\theta \geq \sum_{e \in E} [d_e \cdot (a_e + b_e) + c_e \cdot (f'_e - x_e)] . \quad (6.5)$$

La Figura 6.2 ilustra la solución al  $L$ -TSP-D para la instancia del ejemplo en la Figura 1.1.

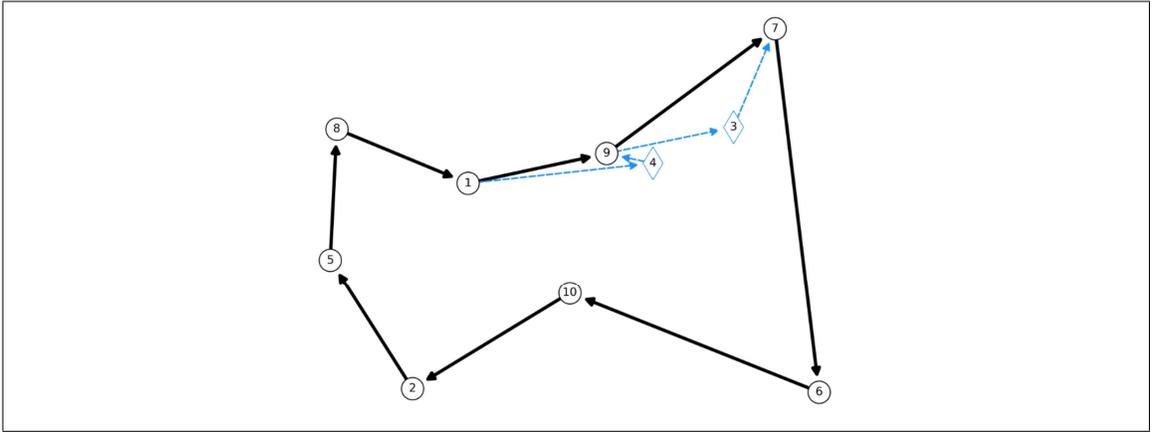


Figura 6.2. Solución óptima al  $L$ -TSP-D

### 6.3. TSP-D con $p$ drones

Otra extensión al TSP-D, denominada  $p$ TSP-D, considera la posibilidad de que el camión transporte  $p \geq 2$  drones desplegables, los cuales pueden atender clientes simultáneamente. Para abordar este caso, se modifica la formulación de  $\mathbb{X}$  reemplazando (3.2e) por

$$\sum_{i \in N} \gamma_i \geq \left\lceil \frac{n}{1+p} \right\rceil \quad (6.6)$$

y agregando

$$\sum_{i \in N \setminus \{1\}} (1 - \gamma_i) \geq p . \quad (6.7)$$

(6.6) exige que el camión visite, al menos, tantos nodos como cada uno de los *drones* y (6.7) fuerza, sin pérdida de optimalidad, que cada *drone* se utilice, al menos, una vez. Esto es resultado del Corolario 1.

En la formulación de  $\mathbb{Y}(x)$  se agrega la variable  $t_i \in \mathbb{R}_+$ , que registra el tiempo acumulado del camión hasta el nodo  $i \in N$ , e  $I_i \in \{0, \dots, k\}$ , que toma como valor la cantidad de *drones* que carga el camión al llegar al nodo  $i \in N$ . Además, se reemplaza (3.3n) por

$$o_{ikj} \leq p \cdot y_{ij}, \quad (i, k, j) \in O \quad (6.8)$$

y se agregan las restricciones

$$t_i \geq v_i \quad i \in N \quad (6.9a)$$

$$t_j \geq t_i + d_{ij} - M \cdot \left( 1 - \sum_{k \in N: (i,j,k) \in O} o_{ijk} - \sum_{k \in N: (k,i,j) \in O} o_{kij} \right) \quad (i, j) \in E \quad (6.9b)$$

$$I_1 = p \quad (6.9c)$$

$$I_i + \left( \sum_{e \in \delta_O^-(j)} o_e - \sum_{e \in \delta_O^+(i)} o_e \right) - p \cdot (1 - x_{ij}) \leq I_j \quad (i, j) \in E \quad (6.9d)$$

$$I_i + \left( \sum_{e \in \delta_O^-(j)} o_e - \sum_{e \in \delta_O^+(i)} o_e \right) + p \cdot (1 - x_{ij}) \geq I_j. \quad (i, j) \in E \quad (6.9e)$$

(6.8) cumple la misma función que (3.3n) y fuerza a que no puedan entrar o salir más de  $p$  *drones* de un nodo. (6.9a) y (6.9b) definen el tiempo acumulado hasta el nodo  $i$  como el máximo tiempo acumulado entre los predecesores de  $i$  más la duración del arco que los une. Finalmente, (6.9c)-(6.9e) conservan el inventario de *drones* en el nodo  $i$ , sumando los *drones* que posee el camión y que recibe en  $i$ , y restando los *drones* desplegados en el nodo antecesor a  $i$ .

En este caso, la duración total de la ruta es equivalente al tiempo acumulado hasta la vuelta al *depot*, por lo que se cambia (5.3a) por

$$\min t_{n+1}$$

La formulación de  $\mathbb{W}(\gamma)$  se modifica usando las variables  $a_e^m$  y  $b_e^m$ , cuya función es la misma que la presentada en (4.6), pero para la ruta del *drone*  $m \in [p] \triangleq \{1, \dots, p\}$ . Así, se reemplaza (4.6) por

$$s_e^m = a_e^m + b_e^m \quad e \in E, m \in [p] \quad (6.10a)$$

$$\sum_{m \in [p]} \sum_{e \in \delta^-(i)} a_e^m = 1 - \gamma_i \quad i \in N \quad (6.10b)$$

$$\sum_{e \in \delta^-(i)} a_e^m = \sum_{e \in \delta^+(i)} b_e^m \quad i \in N, m \in [p] \quad (6.10c)$$

$$\sum_{m \in [p]} \sum_{e \in \delta^+(i)} a_e^m \leq \gamma_i \quad i \in N \quad (6.10d)$$

$$\sum_{m \in [p]} \sum_{e \in \delta^-(i)} b_e^m \leq \gamma_i \quad i \in N \quad (6.10e)$$

$$\sum_{e \in \delta^+(1)} a_e^m = 1 \quad m \in [p] \quad (6.10f)$$

$$\gamma_i^m = \sum_{e \in \delta^+(i)} s_e^m \quad i \in N, m \in [p] \quad (6.10g)$$

$$\sum_{e \in \delta^+(S)} s_e^m \geq \max_{i \in S} \{\gamma_i^m\} \quad \emptyset \neq S \subset N, m \in [p] \quad (6.10h)$$

$$a^m, b^m, s^m \in \{0, 1\}^{|E|} \quad m \in [p] \quad (6.10i)$$

$$\gamma^m \in \{0, 1\}^{|N|} \quad m \in [p] \quad (6.10j)$$

La interpretación de (6.10a), (6.10c)-(6.10h) es la misma que en (4.3a), (4.3c)-(4.3h), donde cada ruta de *drone* las cumple. (6.10b) fuerza a que las visitas en operaciones se realicen por un solo *drone*. Finalmente, (6.10i) y (6.10j) definen la naturaleza de las variables.

Es posible extender la Propiedad 3 a este caso, lo que se expresa en la Propiedad 6. Para ello, considere el conjunto de arcos que forman la ruta óptima del *drone*  $m \in [p]$ , dada una ruta del camión  $(x, \gamma)$ , denominada  $\mathcal{D}_x^m$ .

PROPIEDAD 6. *La duración total de una ruta  $(x, \gamma) \in \mathbb{X}$  en el pTSP-D está acotada inferiormente por la máxima duración de todas las rutas  $\{\mathcal{D}_x^m\}_{m \in [p]}$  y  $\mathcal{R}_x$ , i.e.,*

$$\max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in \mathcal{D}_x^1} d_e, \dots, \sum_{e \in \mathcal{D}_x^p} d_e \right\} \leq \sum_{e \in E} c_e x_e + \sum_{(i,k,j) \in O} w_{ikj}(\mathcal{R}_x). \quad (6.11)$$

DEMOSTRACIÓN. Para cada ruta de *drone* se conserva la Propiedad 2, luego, los arcos en  $\mathcal{D}_x^m$  para  $m \in [p]$  forman operaciones. Desarrollando el lado izquierdo de (6.11), se obtiene que

$$\begin{aligned} \max \left\{ \sum_{e \in E} c_e x_e, \sum_{e \in \mathcal{D}_1} d_e, \dots, \sum_{e \in \mathcal{D}_p} d_e \right\} &= \max \left\{ \sum_{e \in E} c_e x_e, \max_{m \in [p]} \left\{ \sum_{e \in \mathcal{D}_m} d_e \right\} \right\} \\ &= \left( \max_{m \in [p]} \left\{ \sum_{e \in \mathcal{D}_m} d_e \right\} - \sum_{e \in E} c_e x_e \right)_+ + \sum_{e \in E} c_e x_e \\ &\leq \sum_{(i,k,j) \in O} w_{ikj}(\mathcal{R}_x). \end{aligned}$$

La desigualdad se obtiene porque los arcos en  $\mathcal{D}_x^{m^*}$ , donde  $m^* \in \arg \max_{m \in [p]} \left\{ \sum_{e \in \mathcal{D}_x^m} d_e \right\}$ , definen una ruta de *drone* factible y la espera que esta genera es una cota inferior a la espera generada por todas las rutas de *drone* en conjunto, cumpliendo así (4.2). Por ende, se concluye que, para toda ruta del camión  $(x, \gamma) \in \mathbb{X}$ , se cumple (6.11).  $\square$

Utilizando la Propiedad 6 y la Propiedad 4, se reemplaza (5.6) por

$$\left( \sum_{e \in E} d_e s_e^m - c_e x_e \right)_+ \leq Q(x) \leq \theta \quad m \in [p].$$

Análogamente, se puede modificar las formulaciones de  $\mathbb{Y}(x)$  y  $\mathbb{W}(\gamma)$  indexando las variables que construyen las rutas por *drone*. En dicho caso, se utilizan las variables

análogas  $(f^m, o^m)$  y  $s^m$  para el *drone*  $m \in [p]$ . Se mantiene la modificación en la formulación de  $\mathbb{X}$  y se reemplaza (3.3o) y (3.3p) de la formulación de  $\mathbb{Y}(x)$  por

$$\sum_{m \in [p]} \sum_{(i,k,j) \in O} o_{ikj}^m = 1 - \gamma_k \quad k \in N \quad (6.12a)$$

$$\sum_{e \in \delta_O^+(i)} o_e^m + \sum_{e \in \delta^+(i)} f_e^m - \sum_{e \in \delta_O^-(i)} o_e^m - \sum_{e \in \delta^-(i)} f_e^m = \begin{cases} 1 & i = 1 \\ -1 & i = n + 1 \\ 0 & \sim \end{cases} \quad i \in N, m \in [p] \quad (6.12b)$$

$$\sum_{e \in \delta_O^+(i)} o_e^m \leq \gamma_i. \quad i \in N, m \in [p] \quad (6.12c)$$

Este tipo de formulación es conveniente para casos en que  $p$  toma valores pequeños, mientras que la propuesta primeramente para casos en que  $p$  toma valores grandes.

La Figura 6.3 ilustra una solución al 2TSP-D la instancia del ejemplo en la Figura 1.1.

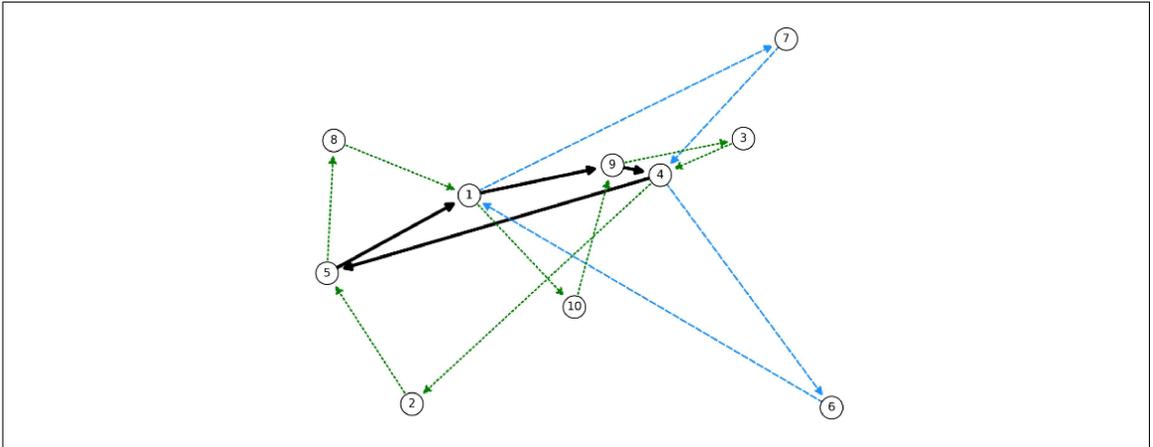


Figura 6.3. Solución al 2TSP-D

## 7. RESULTADOS COMPUTACIONALES

En este capítulo se valida el método propuesto, resolviendo una serie de instancias de prueba simuladas computacionalmente. Los experimentos son ejecutados en un clúster de alto rendimiento, con procesadores Intel E5-2470 de 8 núcleos, 2.5GHz CPU y 32 GB de memoria RAM por instancia. El código es implementado en Python 3.7.3 y los MIPs son resueltos con Gurobi 8.1.1.

### 7.1. Experimentos

Se diseñan 100 instancias de  $n \in \{15, \dots, 25\}$  clientes localizados uniformemente en un cuadrado de lado 100, designado como el área de servicio, y razones de velocidad  $\alpha \in \{1, 2, 4\}$ . Los tiempos de viaje del camión  $c$  se truncan al decimal y se multiplican por 10 para obtener valores enteros, como recomienda Desrochers et al. (1992), y luego se obtiene  $d = \alpha^{-1}c$ . Para resolver (RMP), se prueba, además, reemplazar (3.2b) por restricciones de eliminación de sub-ciclos tipo *Multi-commodity flow* (MCF) (Claus, 1984) (ver Anexo A). La formulación original (3.2) se denomina DFJ, en referencia al artículo seminal de G. Dantzig et al. (1954). Los cortes de optimalidad generan con  $t = t_\gamma^{\max}$ . También, en experimentos preliminares, se testeó la formulación extendida (5.1) que resultó poco competitiva comparada contra el enfoque de descomposición. Adicionalmente, los mejores resultados de la aplicación del método se obtienen para la formulación MCF, por lo que solo se reportan los resultados asociados a esta formulación.

### 7.2. Análisis de resultados

En la Tabla 7.1 se presentan los resultados para 20 instancias por configuración  $(n, \alpha) \in \{15, \dots, 25\} \times \{1, 2, 4\}$ . La columna #Opt presenta la cantidad de instancias resueltas a optimalidad, considerando un tiempo de ejecución de dos horas. La columna Runtime presenta el promedio de los tiempos de ejecución para las instancias resueltas a optimalidad. Finalmente, la columna Gap presenta el promedio del *gap* de optimalidad de las instancias

no resueltas en las dos horas, es decir,

$$\text{Gap} = 100 \cdot \frac{v_{UB} - v_{LB}}{v_{UB}},$$

donde  $v_{UB}$  es el valor objetivo de la solución obtenida y  $v_{LB}$  la mejor cota inferior computada para el valor óptimo.

Empíricamente, se observa que al disminuir  $\alpha$ , el tiempo de ejecución aumenta y disminuye el número de instancias resueltas a optimalidad. También, las soluciones de instancias con valores bajos de  $\alpha$  se tornan más caras. Esto, pues el *drone* se vuelve relativamente más lento y requiere que el camión cubra más nodos en su ruta.

El tiempo requerido para encontrar soluciones con  $\alpha$  bajo es relativamente mayor debido a que las primeras soluciones a (RMP) en el árbol de B&B suelen tener rutas del camión de baja duración, pero cuando  $\alpha$  tiende a 1, el ajuste por demoras del *drone* es mayor. Esto, finalmente exige invertir más tiempo de cómputo buscando soluciones a (RMP) con rutas del camión más largas, cuyo retraso por esperar al *drone* es menor.

Por otro lado, se observa un aumento en el tiempo de ejecución a medida que el tamaño de la instancia crece, pues aumenta el número de soluciones factibles a (RMP) y, por ende, el número de cortes añadidos. El Anexo B presenta un mayor detalle de estos resultados.

Estos resultados son competitivos con los reportados por Roberti & Ruthmair (2019), que logra resolver más instancias a optimalidad, pero es más complejo al requerir B&P. En cambio, el método propuesto puede ser implementado en cualquier *solver* comercial.

La Figura 7.1 presenta una curva acumulada del número de soluciones óptimas en función del tiempo, para todas las instancias con  $n \in \{15, \dots, 20\}$  y  $\alpha = 2$ . Los tiempos calculados corresponden a la diferencia temporal entre el inicio del método y la última actualización de la incumbente. Se observa que más de la mitad de las soluciones se obtienen antes de 30 minutos. Se concluye que es relativamente bajo el número de instancias que aumenta la dispersión y los promedios de los tiempos de cómputo presentados.

Tabla 7.1. Resultados computacionales: TSP-D

$n$	$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)
15	20/20	00:02:39	–	20/20	00:01:21	–	20/20	00:00:08	–
16	20/20	00:04:07	–	20/20	00:02:09	–	20/20	00:00:21	–
17	20/20	00:07:42	–	20/20	00:04:49	–	20/20	00:00:31	–
18	20/20	00:26:47	–	20/20	00:10:37	–	20/20	00:01:14	–
19	20/20	00:31:34	–	20/20	00:14:41	–	20/20	00:01:29	–
20	14/20	00:47:39	1.9	19/20	00:15:43	3.87	20/20	00:02:35	–
21	7/20	00:46:12	3.75	17/20	00:38:55	3.18	20/20	00:06:50	–
22	4/20	01:11:03	5.30	12/20	00:48:14	3.60	19/20	00:07:00	3.87
23	2/20	00:39:12	5.13	5/20	00:55:38	4.13	18/20	00:10:37	3.81
24	1/20	00:50:03	6.77	1/20	00:31:26	5.49	17/20	00:29:17	12.63
25	0/20	–	7.13	0/20	–	6.11	15/20	00:36:41	5.69

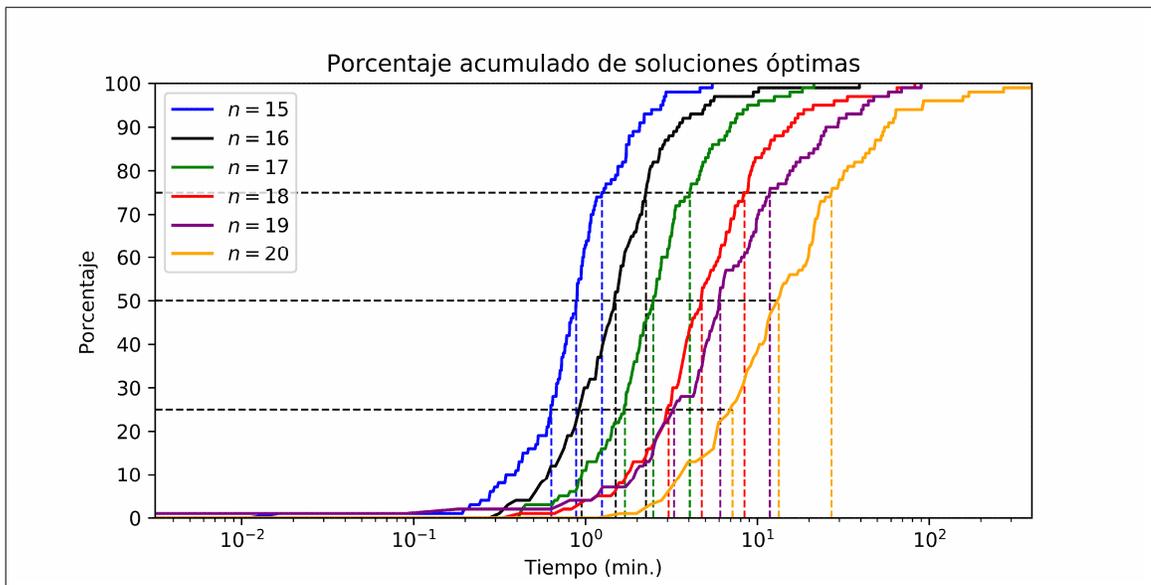


Figura 7.1. Curva acumulada de soluciones óptimas al TSP-D obtenidas en el tiempo.  $\alpha = 2$ . Escala logarítmica

### 7.2.1. Resultados para el TSP-D- $\ell$

Para validar la implementación del TSP-D- $\ell$ , se resolvieron las mismas 20 instancias generadas para resolver el TSP-D. Se testeó, además, la resolución del problema relajando (6.2k) de (RMP), obteniendo menores tiempos de cómputo promedio. Así, se considera esta relajación en los resultados presentados en esta sección.

La Tabla 7.2 resume los resultados del experimento para el TSP-D- $\ell$  en el mismo formato presentado para el TSP-D. En comparación con los resultados del TSP-D, es posible notar un aumento en los tiempos de cómputo promedio y una disminución en el número promedio de soluciones óptimas obtenidas. Esto ocurre especialmente cuando  $\alpha$  tiende a 1. También genera un mayor número de soluciones factibles a (RMP) debido a la relajación de (3.2e). Esto impacta en que el método propuesto tenga un peor desempeño en comparación a su aplicación al TSP-D.

La Figura 7.2 detalla la diferencia entre el número de soluciones óptimas alcanzadas para el TSP-D y el TSP-D- $\ell$ . Empíricamente, se concluye que, en general, el TSP-D- $\ell$  resulta más difícil de resolver con el método propuesto que el TSP-D.

El método propuesto por Bouman et al. (2018) para el TSP-D- $\ell$  reporta soluciones para instancias de hasta 16 nodos con  $\alpha = 2$ . Para instancias de mayor tamaño, sus tiempos de cómputo promedio superan las 12 horas. Si bien las instancias utilizadas en este trabajo no son las mismas, sí son generadas de la misma forma. Comparado con Bouman et al. (2018), el método propuesto en este trabajo brinda una mejora en la capacidad de resolución, detallada en la Tabla B.2 de los Anexos.

A medida que  $\alpha$  aumenta, se espera una mayor diferencia entre la duración de la solución al TSP-D y la duración de la solución al TSP-D- $\ell$ , debido a la restricción impuesta por el supuesto (iii).

La Figura 7.3a detalla la duración promedio de las soluciones a ambos problemas, para 20 instancias nuevas generadas de tamaño  $n = 15$  y razones de velocidad  $\alpha \in$

Tabla 7.2. Resultados computacionales para el TSP-D- $\ell$

$n$	$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)
15	20/20	00:16:57	–	20/20	00:02:57	–	20/20	00:00:57	–
16	19/20	00:26:18	3.84	20/20	00:07:07	–	20/20	00:02:11	–
17	14/20	00:35:43	4.30	19/20	00:15:35	6.47	20/20	00:05:59	–
18	9/20	01:01:02	5.38	19/20	00:26:30	8.95	20/20	00:13:59	–
19	3/20	00:36:32	6.41	15/20	00:38:07	5.93	20/20	00:34:17	–
20	2/20	00:42:06	8.42	11/20	00:57:29	5.29	17/20	00:57:21	6.57
21	1/20	00:50:45	8.44	5/20	01:15:08	8.07	10/20	01:16:45	7.92
22	1/20	01:32:16	10.28	1/20	01:29:37	9.32	1/20	01:43:08	8.65
23	0/20	–	10.91	0/20	–	10.15	1/20	01:20:20	11.16
24	0/20	–	11.53	0/20	–	11.46	0/20	–	12.88
25	0/20	–	11.78	0/20	–	12.17	0/20	–	14.81

		$n$										
		15	16	17	18	19	20	21	22	23	24	25
$\vartheta$	1	0	1	6	11	17	12	6	3	2	1	0
	2	0	0	1	1	5	8	12	11	5	1	0
	4	0	0	0	0	0	3	10	18	17	17	15

Figura 7.2. Diferencia en el número de soluciones óptimas alcanzadas entre el TSP-D y el TSP-D- $\ell$

$\{2.5, \dots, 20\}$ . Se observa que, a medida que  $\alpha$  aumenta, la diferencia en la duración promedio de las soluciones a ambos problemas también lo hace. La causa de esto se aprecia en la Figura 7.3b, donde, a medida que  $\alpha$  aumenta, el número promedio de nodos en la ruta del camión de las soluciones al TSP-D- $\ell$  decrece, mientras que en el caso de las soluciones al TSP-D se mantiene en 8. Este estancamiento del número de nodos en la ruta del camión es producto del supuesto (iii).

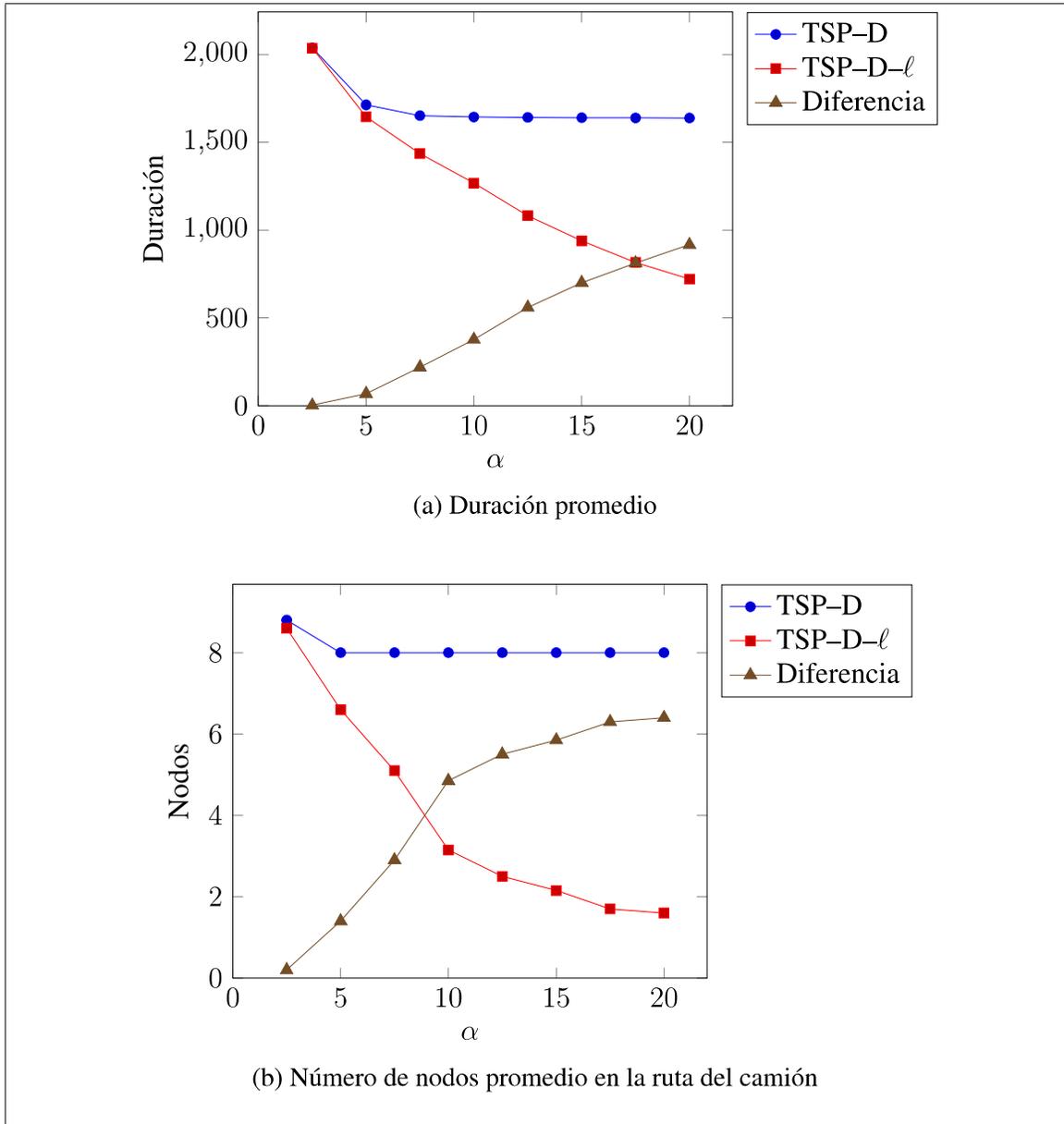


Figura 7.3. Comparación entre soluciones para el TSP-D y para el TSP-D- $\ell$

### 7.2.2. Resultados para el L-TSP-D

También, se limita la autonomía de vuelo del *drone* a  $L \in \{200, 600, 1000\}$  y se resuleven las 20 instancias con  $\alpha = 2$ . La Tabla 7.3 presenta los resultados promedio de este experimento y la Figura 7.4 ilustra la diferencia entre el número promedio de soluciones óptimas obtenidas para el TSP-D y para el L-TSP-D, con  $\alpha = 2$ . Empíricamente, se

observa que la resolución del  $L$ -TSP-D es más rápida que la del TSP-D cuando  $L$  es bajo, lo que se refleja en un mayor número promedio de soluciones óptimas alcanzadas. A medida que  $L$  crece, la resolución tiende a ser similar a la del TSP-D.

Tabla 7.3. Resultados computacionales para el  $L$ -TSP-D

$n$	$L = 200$			$L = 600$			$L = 1000$		
	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)
15	20/20	00:00:00	–	20/20	00:03:50	–	20/20	00:01:29	–
16	20/20	00:00:01	–	19/20	00:02:23	5.54	20/20	00:02:55	–
17	20/20	00:00:02	–	19/20	00:04:15	15.18	20/20	00:05:32	–
18	20/20	00:00:04	–	19/20	00:07:18	9.98	20/20	00:10:34	–
19	20/20	00:00:06	–	18/20	00:10:29	1.23	20/20	00:13:28	–
20	20/20	00:00:13	–	18/20	00:20:38	7.10	19/20	00:21:25	3.60
21	20/20	00:00:21	–	16/20	00:48:26	3.07	16/20	00:32:46	3.07
22	20/20	00:00:39	–	9/20	00:56:50	4.37	12/20	00:57:32	3.39
23	20/20	00:01:57	–	4/20	00:59:59	5.12	5/20	00:55:58	4.37
24	20/20	00:02:49	–	2/20	01:41:57	6.86	1/20	00:36:10	5.61
25	20/20	00:05:06	–	0/20	–	7.45	0/20	–	7.15

$L$	$n$										
	15	16	17	18	19	20	21	22	23	24	25
200	0	0	0	0	0	-1	-5	-9	-16	-19	-20
600	0	1	1	1	2	1	-1	2	0	-1	0
1,000	0	0	0	0	0	0	-1	-1	-1	0	0

Figura 7.4. Diferencia en el número promedio de soluciones óptimas obtenidas entre el TSP-D y el  $L$ -TSP-D

### 7.2.3. Resultados para el $p$ TSP-D

Finalmente, se implementa el problema con dos *drones*, *i.e.*,  $p = 2$ , sobre las mismas 20 instancias generadas para resolver el TSP-D, con  $(n, \alpha) \in \{15, \dots, 20\} \times \{1, 2, 4\}$ . La Tabla 7.4 presenta los resultados promedio de este experimento, que detalla que, a

medida que  $\alpha$  disminuye, el número promedio de soluciones óptimas obtenidas decrece. Los resultados promedio para  $\alpha = 1$  se presentan para las instancias de tamaño  $n \leq 17$ , pues no se logra alcanzar soluciones óptimas para  $n \geq 18$ .

La Figura 7.5 ilustra la diferencia entre el número promedio de soluciones óptimas obtenidas para el TSP-D y para el 2TSP-D. Se observa que el problema con múltiples *drones* es más complejo y difícil de resolver que el TSP-D.

Tabla 7.4. Resultados computacionales para el 2TSP-D

$n$	$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)	#Opt	Runtime	Gap (%)
15	0/20	–	13.68	20/20	00:11:52	–	20/20	00:01:56	–
16	0/20	–	29.54	18/20	00:43:00	6.24	20/20	00:10:28	–
17	0/20	–	55.20	8/20	01:11:02	12.22	19/20	00:15:24	6.41
18				3/20	01:14:14	18.72	17/20	00:19:23	10.22
19				0/20	–	40.17	13/20	00:34:41	40.08
20				0/20	–	56.47	8/20	00:52:25	52.95

		$n$					
		15	16	17	18	19	20
$\alpha$	1	20	20	20	20	20	11
	2	0	2	12	17	20	19
	4	0	0	1	3	7	12

Figura 7.5. Diferencia en el número promedio de soluciones óptimas obtenidas entre el TSP-D y el  $p$ TSP-D

## 8. CONCLUSIONES

En este trabajo se presenta e implementa un método exacto basado en descomposición de Benders, junto a un MIP diseñado acorde para resolver el TSP-D. También, se establecen propiedades y cortes que permiten reducir el tiempo requerido para obtener una solución óptima. El rendimiento del método se estudia en una serie de experimentos computacionales simulados para el TSP-D y sus extensiones. Como resultado general, se observa un aumento en los tiempos de cómputo promedio y en el número promedio de soluciones óptimas obtenidas a medida que aumenta el tamaño de las instancias y disminuye la velocidad relativa del *drone*.

Se proponen tres extensiones al TSP-D: TSP-D admitiendo *loops*, TSP-D con restricción de autonomía de vuelo del *drone* y TSP-D con múltiples *drones*. Junto a ello, se adapta el método propuesto para abordar cada una de estas..

Los beneficios asociados a resolver el TSP-D utilizando descomposición de Benders son:

- (i) Permite aprovechar la estructura de una solución del problema descomponible en una ruta del camión y otra del *drone*.
- (ii) Permite derivar propiedades del conjunto de soluciones óptimas que reducen el espacio de búsqueda.
- (iii) Brinda la flexibilidad necesaria para adaptarse a distintas extensiones del TSP-D.

Finalmente, como trabajo futuro se propone la adaptación del método para abordar el VRP-D. Por ejemplo, si se utiliza para encontrar rutas híbridas de camión y *drone*, se podría considerar dentro de un esquema de B&P y generación de columnas. También, se propone la reformulación e implementación necesaria para abordar la extensión al TSP-D en la cual el camión es capaz de visitar nodos. Existen instancias, como la ejemplificada en la Figura 8.1, en las que si el camión puede visitar nodos, se obtienen soluciones óptimas más cortas. Por último, se propone la reformulación e implementación necesaria

para abordar la extensión al TSP-D en la en la que el *drone* puede visitar más de un nodo por operación.

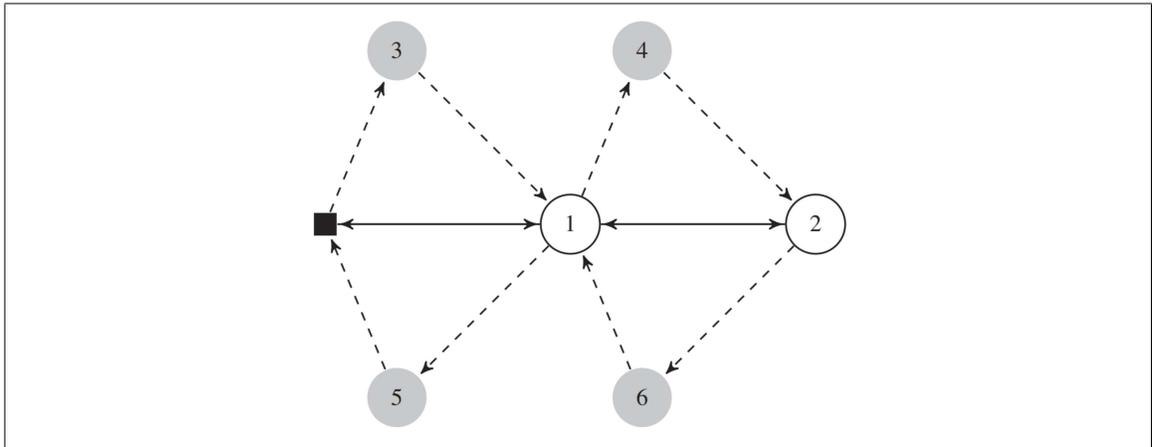


Figura 8.1. Ejemplo de instancia en la que resulta beneficioso poder visitar nodos con el camión. Fuente: Agatz et al. (2018)

## REFERENCIAS

- Adams, E. (2019). *UPS drones are now moving blood samples over North Carolina*. <https://www.wired.com/story/ups-matternet-drone-delivery-north-carolina/>. (Revisado el 01-08-2019)
- Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4), 965–981.
- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. USA: Prentice-Hall, Inc.
- Angulo, G., Ahmed, S., & Dey, S. S. (2016). Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28, 483-499.
- Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The traveling salesman problem: A computational study*. Princeton university press.
- Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, 19(6), 621–636.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1), 238–252.
- bin Othman, M. S., Shurbevski, A., Karuno, Y., & Nagamochi, H. (2017). Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route. *Journal of Information Processing*, 25, 655–666.
- Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4), 528–542.
- Carlsson, J. G., & Song, S. (2017). Coordinated logistics with a truck and a drone. *Management Science*, 64(9), 4052–4069.

- Chao, I.-M. (2002). A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29(1), 33–51.
- Claus, A. (1984). A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic Discrete Methods*, 5(1), 21–25.
- Daimler. (s.f.). *The Mercedes-Benz vision van. for a highly efficient logistics concept.* <https://www.daimler.com/innovation/specials/vision-van/en/>. (Revisado el: 05-08-2019)
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4), 393–410.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80–91.
- de Freitas, J. C., & Penna, P. H. V. (2020). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, 27(1), 267–290.
- Dell'Amico, M., Montemanni, R., & Novellani, S. (2019). Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. *arXiv preprint arXiv:1905.13463*.
- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2), 342–354.
- Drexl, M. (2012). Synchronization in vehicle routing – a survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3), 297–316.
- Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., & Rich, R. (2016). Optimization of a truck–drone in tandem delivery network using k–means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM)*, 9(2), 374–388.

- Gambella, C., Lodi, A., & Vigo, D. (2017, 09). Exact solutions for the carrier–vehicle traveling salesman problem. *Transportation Science*, *52*. doi: 10.1287/trsc.2017.0771
- Garone, E., Determe, J.-F., & Naldi, R. (2014). Generalized traveling salesman problem for carrier–vehicle systems. *Journal of guidance, control, and dynamics*, *37*(3), 766–774.
- Garone, E., Naldi, R., & Casavola, A. (2011, 07). A traveling salesman problem for a class of carrier–vehicle systems. *Journal of Guidance, Control, and Dynamics*, *34*. doi: 10.2514/1.50539
- Garone, E., Naldi, R., Casavola, A., & Frazzoli, E. (2008, 01). Cooperative path planning for a class of carrier–vehicle systems. In (p. 2456-2462). doi: 10.1109/CDC.2008.4739357
- Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, *34*(3), 307–318.
- Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min–cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, *86*, 597–621.
- Laporte, G., & Louveaux, F. V. (1993). The integer L–shaped method for stochastic integer programs with complete recourse. *Operations research letters*, *13*(3), 133–142.
- Laporte, G., & Louveaux, F. V. (1998). Solving stochastic routing problems with the integer L–shaped method. In *Fleet management and logistics* (pp. 159–167). Springer.
- Laporte, G., Louveaux, F. V., & Van Hamme, L. (2002). An integer L–shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, *50*(3), 415–423.
- Mathew, N., Smith, S. L., & Waslander, S. L. (2015). Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1298–1308.

- Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4), 326–329.
- Moshref-Javadi, M., Hemmati, A., & Winkenbach, M. (2020). A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Applied Mathematical Modelling*, 80, 290–318.
- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86–109.
- Orphanides, K. (2016). *Drones can now land on moving cars*. <https://www.wired.co.uk/article/drone-lands-on-moving-car>. (Revisado el 14-08-2019)
- Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4), 411–458.
- Poikonen, S., & Golden, B. (2019). The mothership and drone routing problem. *INFORMS Journal on Computing*.
- Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1), 34–43.
- Ponza, A. (2016). *Optimization of drone-assisted parcel delivery* (Unpublished master's thesis). Università Degli Studi di Padova.
- Pugliese, L. D. P., & Guerriero, F. (2017). Last-mile deliveries by using drones and classical vehicles. In *International conference on optimization and decision science* (pp. 557–565).
- Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3), 801–817.

Roberti, R., & Ruthmair, M. (2019). Exact methods for the traveling salesman problem with drone. *Optimization Online*.

Robinson, D. J. (2008). *An introduction to abstract algebra*. Walter de Gruyter.

Rose, C. (2013). *Amazon's jeff bezos looks to the future*. <https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/>. (Revisado el 17-08-2019)

Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6-7), 971–983.

Stewart, J. (2017). *A drone-slinging UPS van delivers the future*. <https://www.wired.com/2017/02/drone-slinging-ups-van-delivers-future/>. (Revisado el 01-08-2019)

Tang, Z., van Hoes, W.-J., & Shaw, P. (2019). A study on the traveling salesman problem with a drone. In *International conference on integration of constraint programming, artificial intelligence, and operations research* (pp. 557–564).

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2), 146–160.

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. SIAM.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9), 797–809.

Ulmer, M., & Thomas, B. (2018, 08). Same-day delivery with a heterogeneous fleet of drones and vehicles. *Networks*. doi: 10.1002/net.21855

Van Dijck, E., Bouman, P., & Spliet, R. (2018). *A branch-and-cut algorithm for the traveling salesman problem with drone* (Unpublished master's thesis). Erasmus University of Rotterdam.

Wang, Z., & Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation research part B: methodological*, 122, 350–364.

Wilke, J. (2019). *A drone program taking flight*. <https://blog.aboutamazon.com/transportation/a-drone-program-taking-flight>. (Revisado el 01-08-2019)

Yurek, E. E., & Ozmutlu, H. C. (2018). A decomposition–based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 91, 249–262.

Zaccara, G. (2019). *Rwanda drone network is potential model for other countries*. <https://pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases&id=1462760058735-442>. (Revisado el 01-08-2019)

## **ANEXOS**

## A. FORMULACIÓN MCF PARA ELIMINACIÓN DE SUB-CICLOS

En este Anexo, se presentan las restricciones tipo *Multi-commodity flow* para la eliminación de sub-ciclos. Se define la variable binaria  $f_e^m \in \{0, 1\}$ , la cual es igual a 1 si y solo si se envía flujo del *commodity*  $m \in N \setminus \{1\}$  por el arco  $e \in E$ . La formulación MCF se construye modificando (3.2e) por

$$f_{ij}^m \leq x_{ij} \quad (i, j) \in E \quad (\text{A.1a})$$

$$\sum_{e \in \delta^+(1)} f_e^m = \gamma_m \quad m \in N \setminus \{1\} \quad (\text{A.1b})$$

$$\sum_{e \in \delta^-(1)} f_e^m = 0 \quad m \in N \setminus \{1\} \quad (\text{A.1c})$$

$$\sum_{e \in \delta^+(i)} f_e^m - \sum_{e \in \delta^-(i)} f_e^m = 0 \quad i, m \in N \setminus \{1\}, i \neq m \quad (\text{A.1d})$$

$$\sum_{e \in \delta^+(m)} f_e^m - \sum_{e \in \delta^-(m)} f_e^m = -\gamma_m \quad m \in N \setminus \{1\} \quad (\text{A.1e})$$

$$f \in \{0, 1\}^{|E \times N|}. \quad (\text{A.1f})$$

## B. RESULTADOS COMPUTACIONALES PARA EL TSP-D Y EL TSP-D- $\ell$

En la Tabla B.1 se presentan los resultados computacionales promedio del método para resolver el TSP-D, sin tiempo de cómputo límite. La columna Runtime contiene los tiempos de ejecución, Explored nodes el número de nodos del árbol de B&B explorados, y Cuts el número total de cortes añadidos. La Tabla B.2 presenta los resultados computacionales promedio para el TSP-D- $\ell$ , con  $\alpha = 2$ , bajo el mismo formato.

Tabla B.1. Resultados computacionales para el TSP-D. Desviación estándar entre paréntesis

$n$	$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
	Runtime	Explored nodes	Cuts	Runtime	Explored nodes	Cuts	Runtime	Explored nodes	Cuts
15	00:02:36 (00:02:20)	135999 (329561)	1860	00:01:28 (00:01:20)	39749 (96048)	679	00:00:09 (00:00:07)	1526 (2171)	101
16	00:05:38 (00:04:50)	165975 (235057)	3069	00:02:18 (00:02:12)	96428 (235277)	1099	00:00:22 (00:00:17)	5003 (2905)	426
17	00:08:31 (00:05:32)	299784 (387475)	3971	00:04:23 (00:05:23)	93621 (98359)	2348	00:00:31 (00:00:39)	15999 (53162)	281
18	00:36:05 (00:28:49)	473012 (392596)	6856	00:11:07 (00:20:48)	136008 (123976)	3385	00:01:17 (00:01:14)	19776 (27712)	952
19	00:34:12 (00:24:35)	682093 (738075)	8815	00:12:23 (00:11:55)	234431 (260969)	3687	00:01:38 (00:01:35)	9176 (8876)	683
20	01:27:16 (01:04:52)	1329044 (1173419)	14165	00:24:40 (00:29:37)	351655 (291821)	5573	00:02:27 (00:02:05)	32777 (31829)	1358

Tabla B.2. Resultados computacionales para el TSP-D- $\ell$ . Desviación estándar entre paréntesis

$n$	Runtime	Explored nodes	Cuts
15	00:03:17 (00:03:43)	56088 (45808)	2070
16	00:07:10 (00:09:22)	104435 (105457)	3654
17	00:31:13 (01:18:43)	294457 (564015)	7577
18	01:06:47 (03:18:03)	405034 (641947)	11933
19	01:56:53 (03:29:39)	626847 (722360)	18600