**A MATH HEURISTIC APPROACH FOR THE AIR-CARGO RECOVERY PROBLEM CONSIDERING DEMAND DISRUPTIONS**

**ABSTRACT (MAX 100 palabras)**

En el transporte aéreo de carga suelen ocurrir cambios inesperados entre la demanda planificada y la que se materializa provocando ajustes en los itinerarios para hacer frente a esta situación. En este trabajo proponemos una math heuristic basada en un enfoque de generación de columnas donde cada subproblema es resuelto utilizando una heurística ad-hoc Nuestro modelo fue testeado en instancias hasta 75 pedidos bajo diferentes niveles de penalidad. Los resultados muestran que en un tiempo límite de 2 horas, la metodología propuesta obtiene ganancias en promedio 33% mayores que al resolver directamente el problema original (MIP) y 9.53% mejor que operar la misma planificación original optimizando únicamente el ruteo de las cargas (FIX route)

*Keywords:  Air Cargo Schedule Recovery; Airline Schedule Recovery; Disruption Management; Air cargo Rescheduling; Pickup and delivery, Column Generation, math heuristic*

**1. INTRODUCCIÓN**

En el transporte aéreo existen dos maneras de transportar la carga, mediante la capacidad disponible en los bellys de aviones de pasajeros, luego de cargar a los pasajeros y sus maletas, o por medio de aviones cargueros exclusivamente dedicados al transporte de carga. Aún cuando el incremento de capacidad de payload en aviones con belly ha casi triplicado la de cargueros en los últimos años (IATA, 2015a), estos presentan ciertas ventajas que lo diferencian a los primeros y los hacen necesarios para la operación de una aerolínea de carga. Entre las ventajas que presentan los cargueros, los cuales son el centro de este trabajo, está la posibilidad de transportar cierto tipo de carga que los aviones tipo belly no pueden, como carga sobredimensionada, sustancias peligrosas, entre otras. Además, la oferta de belly no siempre calza con los requerimientos especiales de la demanda de carga, ya sea por los horarios de vuelo, por la capacidad disponible o porque no siempre los mercados orígenes-destino atractivos para la carga tienen una mayor frecuencia en vuelos de pasajeros (Boeing, 2014).

Una de las principales diferencias entre el transporte aéreo de carga y el de pasajeros es la alta incertidumbre de la demanda a ser transportada en cada itinerario carguero. Esta incertidumbre se explica porque las reservas son realizadas en ventanas de tiempo reducidas, la carga reservada no se presenta o lo hace parcialmente (*No Show*), llega fuera de plazo o a última hora (Wada et al, 2017). Esta incertidumbre es revelada solo pocas horas antes del vuelo generándose un desbalance entre la oferta planificada y las demandas reales. Este desbalance produce pérdidas inevitables para las aerolíneas que se reflejan especialmente en el mercado de carga aérea, donde los aviones cargueros presentan un bajo factor de ocupación (45%) si se compara con el alcanzado por los aviones de pasajeros en 2015 (80,5%) (IATA, 2015b),

Para afrontar estas disrupciones que se producen en la demanda, las cuales son difíciles de anticipar, las aerolíneas realizan ajustes focalizados y de último minuto en la planificación original, basados en la experiencia de los tomadores de decisión, quienes modifican de manera manual los itinerarios. Estas modificaciones consideran medidas como: cancelación de vuelos, reruteo de aviones, volar aviones vacíos para reacomodar la carga (siempre cuando el análisis de rentabilidad lo permita), agregar escalas entre aeropuertos o efectuar vuelos de "ida-vuelta" (*roundtrip*), así como también, una combinación de estas alternativas. Estas acciones, las cuales deben ser tomadas en cortos periodos de tiempo, son de naturaleza correctiva y locales, sin buscar construir un itinerario nuevo desde cero. Sin embargo, en la práctica alteran diferentes etapas en la planificación de itinerarios de aviones cargueros como son el diseño de itinerarios, rotación de aviones, asignación de tripulación y ruteo de la carga. El objetivo de realizar correcciones menores radica en tratar de afectar lo menos posible los turnos ya asignados a las tripulaciones (Crew scheduling). Sin embargo, estas soluciones son por lo general sub óptimas lo que afecta directamente los costos de la compañía.

El diseño de itinerarios en la industria aérea ha sido largamente estudiado en el mercado de pasajeros. Etschmaier y Mathaisel (1985) son los primeros en realizar una revisión del estado del arte al respecto. Desde esa fecha diversos trabajos han abordado el tema de planificación aérea abarcando entre otros el diseño incremental de itinerarios aéreos de pasajeros (Barnhart et al, 2003; Berge y Hopperstad, 1993), el problema simultáneo de diseño de itinerarios y asignación de flota (Levin, 1971; Rexing et al., 2000; Lohatepanont y Barnhart, 2004) y el estudio de las perturbaciones o eventos inesperados el cual ha sido abordado desde dos perspectivas. La primera se basa en modelos operacionales de recuperación de itinerarios (Bratu y Barnhart, 2006; Rosenberger et al, 2003), mientras que la segunda se basa en el desarrollo de modelos robustos de planificación de itinerarios (Gao et al, 2009; Lan et al, 2006; Rosenberger et al, 2004; Pita et al, 2013; Froyland y Maher, 2014).

Aún cuando el estudio de modelos operacionales de recuperación de itinerarios ha sido largamente estudiado en la industria de pasajeros, existen pocos trabajos que aborden este problema en la industria de carga aérea. En este sentido, las perturbaciones en la industria de pasajeros tienen una naturaleza diferente al mercado carguero y vienen dadas principalmente por efectos climáticos o por la propagación de demoras debido a cancelaciones o retrasos de vuelos y no por las fluctuaciones que sufre la demanda. Otra importante diferencia radica en las acciones correctivas que se realizan en caso de perturbaciones. En el caso de pasajeros, los tramos de vuelo no se pueden cambiar en el corto plazo debido a la preferencia de los pasajeros hacia determinados itinerarios. En cambio, en el caso de los cargueros, interesa que la carga llegue a destino en los tiempos estipulados, independiente del itinerario efectuado, lo que da una mayor flexibilidad en el cambio de tramos de vuelos.

En relación al transporte aéreo de carga la literatura es más limitada. Uno de los primeros estudios de esta materia es el realizado por Marsten y Muller (1980), quienes proponen un modelo de programación entera mixta para resolver dos problemas estratégicos para una red hub-and-spoke: el diseño de itinerarios y la selección de flota. Lin y Chen (2003) proponen un Multi-commodity Flow Problem (MCFP) para el diseño de itinerarios en la red de aviones belly, considerando la selección de aeropuertos de transferencia para conectar las redes de carga en Taiwán y China.

Yan et al. (2006) presentan un modelo que integra las etapas estratégicas de diseño de itinerarios y asignación de flota para aviones cargueros considerando un horizonte de planificación semanal, que al ser resuelto en poco tiempo puede ser empleado en operaciones de corto plazo. Los autores formulan el problema como un MCFP considerando flota única. El problema busca integrar una red de vuelos a una red de carga (pares OD), para lo cual consideran todos los tramos de vuelo posibles, a modo de diseñar un itinerario desde cero sin tener en cuenta una planificación base ni costos adicionales por cancelar vuelos. Para su resolución emplean diferentes heurísticas que consideran el número de paradas que debe hacer el avión tras recoger un pedido y llevarlo a su destino. A pesar de lo novedoso de este trabajo, una de las limitaciones del estudio es que el ruteo depende de la heurística utilizada, las cuales restringen el número de paradas que un avión puede realizar tras recoger un pedido. Yan y Chen (2008) extienden el modelo mencionado anteriormente para escenarios donde existen alianzas entre aerolíneas que requieren de modelos coordinados de itinerario. En Tang et al (2008) se formula un modelo que integra el diseño de itinerarios para aviones de pasajeros, cargueros y belly. El problema se formula como un integer MCFP y se propone una serie de heurísticas para su resolución.

Derigs et al (2009) aborda la planificación de itinerarios cargueros mediante una formulación que simultáneamente optimiza la selección de vuelos, la rotación de aviones y el ruteo de la carga. Este enfoque da énfasis a una mejora incremental de itinerarios mediante cambios en la selección de vuelos. Continuando con este análisis, Derigs y Frederichs (2013) hacen una revisión general de la planificación aérea de carga y los subproblemas que esta conlleva, para luego focalizarse en la etapa de diseño de itinerarios cíclicos, donde a partir de una lista de vuelos prioritarios y opcionales selecciona la combinación de vuelos más rentable. Si bien consideran realizar los menores cambios posibles a una planificación base, ésta solo se hace a nivel de vuelos predefinidos los que son generados externamente por los tomadores de decisión.

A partir de todo lo anterior, se observa que el problema de ajustes a itinerarios cargueros, en respuesta a disrupciones de último minuto en la demanda, no ha sido abordado por la literatura a nivel operacional. Este problema tiene características especiales donde el tiempo es un recurso limitado, no se cuenta con una lista de vuelos definidas a priori y se requieren ajustes menores y focalizados en los itinerarios de la planificación base. Los principales aportes de este trabajo que lo diferencia de los estudios presentados anteriormente son:

1. El desarrollo de un modelo que permite reaccionar a disrupciones de la demanda de carga de corto plazo. Esto no ha sido tratado en el transporte aéreo de carga, y en el transporte de pasajeros los modelos reactivos responden a disrupciones que son de naturaleza diferente a la demanda (clima, cancelaciones y demoras de vuelos).
2. Este modelo reactivo considera de manera simultánea la modificación de un itinerario base existente, el ruteo de aviones, el ruteo de la carga y trata de impactar lo menos posible a las tripulaciones afectadas, mediante la incorporación de costos de penalización por cambios en la planificación base. Esto impone restricciones que no tienen los problemas de planificación que en general elaboran itinerarios cíclicos, como por ejemplo que los aviones deben empezar y terminar en aeropuertos específicos, en periodos de tiempo determinados.

El resto del paper se divide en las siguientes secciones. La Sección 2 describe la modelación matemática del problema incluyendo los principales supuestos y la notación usada. En la sección 3, presentamos el método de solución incluyendo la descripción del método de generación de columnas y la heurística desarrollada para resolver cada subproblema. Los resultados computacionales se reportan en la sección 4 seguido de las conclusiones en la sección 5.

**2. MODELACIÓN**

El problema presentado se formula como una variante al problema de Pickup and Delivery con Ventanas de Tiempo (PDPTW) (Dumas et al., 1991; Cordeau et al., 2002; Parragh et al., 2008), con ajustes y restricciones adicionales para el transporte aéreo de carga. En este caso, los pedidos son continuamente divisibles, de manera que los aviones pueden recoger cargas parciales de un pedido y un pedido puede ser transportado por más de un avión. En la red, cada nodo corresponde a un evento que representa ya sea un aterrizaje, despegue o estadía en un determinado aeropuerto con su correspondiente carga o descarga de un pedido. Un pedido se define por un nodo pickup donde se carga el requerimiento de transporte y un nodo delivery donde se descarga, cada uno ubicado en un aeropuerto, con su respectiva ventana de tiempo y duración de servicio.

Los arcos que conectan a estos nodos representan arcos terrestres (estadía en un aeropuerto), si ambos nodos pertenecen a un mismo aeropuerto, o tramos de vuelo, si estos se encuentran en diferentes aeropuertos. Por otro lado, para cada avión se establece una subred en la cual puede operar, debido a restricciones operacionales o a derechos de vuelo que impiden la operación libre en todos los aeropuertos de la red. A su vez, se considera una flota heterogénea, donde cada avión tiene asociada una capacidad de transporte de carga, una ubicación inicial y final para el horizonte temporal y una planificación base a la cual se le deben hacer los menores ajustes posibles.

A continuación, se presentan los principales supuestos de modelación, la notación del modelo y su formulación matemática, para luego introducir el método de descomposición usado para resolverlo.

**2.1. Supuestos**

The proposed model considers the following assumptions:

1. Los tiempos de carga y descarga son determinísticos y específicos a cada tipo de pedido que se transporta.
2. No se considera la transferencia de carga entre aviones cargueros.
3. Los pedidos son continuamente divisibles, de manera que los aviones pueden recoger cargas parciales de un pedido y un pedido puede ser transportado por más de un avión.
4. Las restricciones de capacidad están asociadas a las cargas en toneladas que se transportan lo que es consistente con la práctica en la industria.
5. No se consideran penalidades por dejar pedidos de la planificación base, parcial o completamente, abajo del avión.

**2.2. Notación**

**Conjuntos**

|  |  |  |
| --- | --- | --- |
| *K* | : | Conjunto de aviones identificados por su matrícula*,* |
| *P* | : | Conjunto de pedidos, . Sea , entonces el nodo *pickup* del pedido *i* es y su nodo *delivery .* |
|  | : | Conjunto de pedidos que pueden ser llevados por el avión *k*, . |
|  | : | Conjunto de nodos *pickup* y *delivery* que pueden ser visitados por el avión *k*, junto con los nodos de ubicación inicial y final al término del horizonte de planificación,{}. |
|  | : | Conjunto de arcos que pueden ser recorridos por el avión *k*, , con . Los arcos que conectan nodos *pickup* o *delivery* con la ubicación inicial y final de los aviones, definidos como () y (), son arcos ficticios que no representan un transporte de carga, solo permite asegurar que las rutas terminen en los aeropuertos requeridos para cada avión. |
|  | : | Conjunto de nodos antecesores a *i* para el avión *k*, |
|  | : | Conjunto de nodos sucesores a *i* para el avión *k*, |
|  | : | Conjunto de macro-nodos. Los macro-nodos permiten consolidar nodos pickup y delivery consecutivos de un mismo aeropuerto, de modo tal que un macronodo define un nodo de origen o destino de un tramo de vuelo en la planificación base. |
|  | : | Conjunto de nodos *pickup* y/o *delivery* que componen el macro-nodo . |
|  | : | Conjunto de tramos de vuelo en la planificación base para el avión *k*, indexado como con . |

**Parámetros**

|  |  |  |
| --- | --- | --- |
|  | : | Costo fijo de recorrer el arco/vuelo por el avión [US$/arco]. |
|  | : | Costo de penalización por modificación del vuelo para avión . Este costo puede representar el tener que disponer de una tripulación extra o tener que trasladar una tripulación desde otro aeropuerto, entre otros posibles costos [US$/vuelo]. |
|  | : | Tarifa por transportar el pedido [US$/tonelada]. |
|  | : | Tiempo de viaje entre el nodo para el avión *k* [hrs]. |
|  | : | Tiempo de carga o descarga en el nodo [hrs]. |
|  | : | Instante de tiempo mínimo en que el nodo puede ser visitado [hrs]. |
|  | : | Instante de tiempo máximo en que el nodo puede ser visitado [hrs]. |
|  | : | Carga total demandada por el pedido [toneladas]. |
|  | : | Capacidad del avión [toneladas]. |

**Variables de decisión**

|  |  |  |
| --- | --- | --- |
|  | : | Variable binaria que toma valor 1 si el avión *k* recorre el arco , 0 en otro caso. |
|  | : | Instante de tiempo en que el avión *k* inicia su visita al nodo *i*. [hrs] |
|  | : | Cantidad de carga que se asigna al nodo *i.* Representa la carga que se recoge () si el nodo *i* es *pickup* y la carga que se entrega () si el nodo *i* es *delivery* [toneladas]. |
|  | : | Carga que lleva el avión *k* al salir del nodo [toneladas]. |
|  | : | Variable auxiliar que toma valor 1 si el tramo de vuelo se cancela y 0 si no. |

**2.3. Modelo matemático**

A continuación se presenta la modelación matemática del problema.



El objetivo del problema (1) corresponde a la maximización de los ingresos por llevar cada pedido menos los costos operacionales y de penalización por modificación del vuelo de la planificación base. Las restricciones (2) y (3) en conjunto determinan que la penalización de un vuelo se gatilla ya sea por una cancelación del vuelo de la planificación base o que el vuelo sea operado por un avión distinto. Las restricciones (4) y (5) restringe a que cada avión *k* comience (termine) en el nodo de inicio (fin) del horizonte de planificación. La restricción (6) asegura la continuidad entre nodos consecutivos de una ruta factible. La restricción (7) establece el instante de llegada a cada nodo. Las restricciones (8), (9), (10) y (11) aluden a las restricciones de capacidad asociadas al avión *k*. La restricción (12) limita la cantidad máxima que puede ser recogida en un determinado nodo pickup. Con la restricción (13) se asegura que la carga recogida en un nodo pickup sea de igual tamaño que la carga entregada en su respectivo nodo delivery. En (14) se restringe que si un avión recorre un nodo pickup entonces debe visitar el nodo delivery de ese pedido. Con (15) se asegura que el nodo pickup sea visitado antes que el nodo delivery. La restricción (16) limita los instantes de arribo a cada nodo en base a sus ventanas de tiempo. Finalmente, (17)-(18) establece la naturaleza de las variables.

El problema anterior es no lineal en el conjunto de restricciones (7), (8) y (10). La no linealidad en (7) y (8) se corrige en base a la introducción de parámetros big-M propuestos por Cordeau (2006) y la restricción (10) puede fácilmente reemplazarse por un conjunto de restricciones lineales.

Aunque la tripulación no se modela directamente como restricción adicional al problema, ésta se considera implícitamente en el objetivo de realizar los menores cambios posibles a la planificación base, mediante la minimización de los costos totales de penalización (tercer componente de la función objetivo).

El problema planteado es *NP-hard* (Savelsbergh y Sol, 1995), esto implica que resolverlo mediante solución exacta sea altamente costoso en términos de tiempo computacional. En el caso de nuestro problema el obtener una solución exacta, pero en un periodo extenso de tiempo, puede significar que sea imposible reaccionar de manera oportuna a los cambios que se producen a último momento en la demanda de carga.

**3. MÉTODO DE SOLUCIÓN**

Para resolver el problema de PDPTW se han propuestos diversos métodos, destancando métodos exactos (Sigurd et al, 2004, Ropke et al, 2007; Ropke y Cordeau, 2009) y metaheurísticos (Ropke y Pisinger, 2006; Qu y Bard, 2012; Cherkesly et al, 2015). Nosotros proponemos combinar ambos métodos mediante una math-heuristic algorithm, debido al uso de heurísticas en conjunto con técnicas de modelos de programación matemática (Maniezzo et al, 2010). En particular, el problema presenta una estructura de bloques, que permite emplear Descomposición de Dantzig-Wolfe (Dantzig y Wolfe, 1960) para su resolución. La descomposición de Dantzig Wolfe consiste en este caso, en dividir la formulación en un Problema Maestro Restringido (PMr) y un conjunto de K-Subproblemas (K-SPs), donde cada uno genera rutas factibles para un avión en particular. La relajación lineal del PMr se resuelve en base a un enfoque de Generación de Columnas (Wolsey, 1998; Barnhart et al, 1998; Desrosiers y Lübbecke, 2005; Feillet, 2010) mientras que los *k-*Subproblemas se resuelven bajo un enfoque heurístico. Para obtener finalmente la solución entera, se utiliza el mismo enfoque empleado por Xu at al (2003), que consiste en resolver una versión restringida del PMr el cual incluye solamente aquellas columnas generadas al resolver la relajación lineal.

**3.1 General Framework**

A continuación se presenta la formulación del Problema Maestro Restringido y el *k* Subproblema.

**3.1.1 Problema Maestro Restringido (PMr)**

Se definen los siguientes conjuntos, parámetros y variables adicionales:

**Conjuntos**

|  |  |  |
| --- | --- | --- |
|  | : | Conjunto de rutas factibles para el avión *k.* |
|  | : | Conjunto de nodos visitados por la ruta *r*. |
|  | : | Conjunto de pedidos que forman parte de la ruta *r*. |

**Parámetros**

|  |  |  |
| --- | --- | --- |
|  | : | Representa la ruta generada por el *k*-Subproblema. Vector que se define como sigue: |
|  |  |  |
|  | : | Utilidad asociada a la ruta *r* del avión *k*. Vector cuyos elementos corresponden a los coeficientes asociados a cada familia de variables y se define como sigue: |
|  |  |  |
|  |  | De este modo, el producto punto corresponde al valor del término de la función objetivo original para el avión *k* evaluada en la solución o ruta generada por el *k*-Subproblema. |
|  | : | Carga del nodo *pickup* asignada al avión *k* en la ruta *r*, forma parte de la familia en el vector . |

**Variables**

|  |  |  |
| --- | --- | --- |
|  | : | Variable binaria que toma valor 1 si la ruta *r* se asigna al avión *k*. |

De esta forma, el PMr puede ser formulado como sigue:



El objetivo de este problema (19) puede ser descrito como una combinación lineal convexa de los valores de las soluciones o rutas generadas por los subproblemas. De esta forma, las soluciones de los subproblemas pueden ser usadas para generar una solución factible del problema original. La restricción (20) es vinculante a todos los subproblemas y corresponde a la restricción de demanda para cada nodo pickup del pedido , relacionada con la restricción (12) del problema original. La restricción (21) es conocida como Restricción de Convexidad y se puede interpretar como una restricción que asegura que a cada avión *k* se le asigne solo una ruta *r* del conjunto de rutas factibles a operar por ese avión. La restricción (22) corresponde a la naturaleza binaria de la variable de decisión, la que debe ser relajada para facilitar la resolución del problema y así poder obtener las valores duales que se ingresan a los K-Subproblemas.

Cada vez que se resuelve el PMr relajado, se obtienen los valores duales para cada nodo pickup de la red y para cada *k*-Subproblema. Cada se interpreta como el costo de oportunidad de llevar una unidad adicional del pedido *i* antes que el resto de los pedidos. Mientras que se interpreta como la utilidad de la ruta óptima generada hasta ese momento para el avión *k*.

**3.1.2** ***k*-Subproblema (*k*-SP)**

Cada problema satélite *k*-SP tiene como finalidad encontrar la ruta óptima para un avión *k* en particular y está formado por las restricciones no vinculantes del problema original. El problema *k*-SP se puede formular de la siguiente manera:



La función objetivo (23) en este caso representa los costos reducidos ( del problema original. La solución obtenida por este problema es una ruta factible para el avión *k*, sin considerar las restricciones de demanda. Si el valor de la función objetivo o costo reducido de esta solución es mayor a 0, sería una ruta beneficiosa para el problema y conviene incorporarla al Problema Maestro Restringido.

Debido a que resolver el PMr de manera exacta involucra un número exponencial de columnas (todas las rutas factibles por cada vehículo), se propone resolverlo bajo un enfoque de generación de columnas, donde el PMr considera un número acotado de columnas iniciales y luego se incorporan nuevas a partir de la resolución heurística de cada *k-*SP. Si al finalizar la ejecución de la heurística para un *k*-SP la ruta final generada tiene costo reducido positivo, esta columna se agrega al PMr y se descarta en caso contrario. El algoritmo de generación de columnas itera hasta que no sea posible encontrar subproblemas que hayan generado columnas con costo reducido positivo. Una vez finalizada la generación de columnas, se resuelve el PMr (imponiendo la restricción de naturaleza binaria de las variables) sobre una versión restringida del problema que incluye únicamente las columnas generadas hasta ese momento. Un resumen de la metodología de solución se muestra en la Figura 1



Figura 1: Esquema del algoritmo de resolución

Como en la primera iteración, el algoritmo aún no ha incorporado columnas al PMr, resulta necesario generar un conjunto de columnas iniciales. Para esto se generan tres conjuntos de columnas para cada avión:

1. Planificación base: El primer conjunto de columnas corresponde a la planificación base de cada avión y a modificaciones de ésta luego de aplicar los operadores de local search, drop and add y swap respectivamente, los que serán descritos en la sección 3.2.2. Si la planificación inicial no es factible después de las disrupciones, esta se factibiliza, de ser necesario, ajustando las ventanas de tiempo y las cargas.
2. Dead-heading: Consiste en la posibilidad de que un avión se dirija directamente desde su aeropuerto inicial al final (e.g. []) sin servir ningún pedido.
3. Heurística constructiva: el último conjunto de columnas es generada a partir de una heurística constructiva y a modificaciones de éste luego de aplicar los mismos operadores de local search usados para modificar la planificación base. La heurística constructiva es la siguiente:

La heurística constructiva, va construyendo rutas para todos los aviones de manera paralela de forma tal de generar rutas más balanceadas (Labadie et al, 2016). Esto es, agrega primero un pedido a cada una de las rutas, luego un segundo y así sucesivamente hasta servir todos los pedidos. Cada nuevo pedido *i*, se agrega al final de la ruta visitando su nodo de pickup *i+* seguido inmediatamente por su nodo de delivery *i-*. El nuevo pedido a insertar en la ruta corresponderá a aquel que represente el menor costo de inserción. Sea *j* el último pedido insertado en la ruta *k* hasta ese momento y *Nk* el conjunto de pedidos pendientes y que pueden ser transportados por el avión *k*. Luego el próximo pedido a insertar en la ruta será aquel que cumpla con y satisfaga las restricciones de ventanas de tiempo y capacidad. Al finalizar este proceso se tendrá, para cada avión *k*, una ruta del tipo . Notar que este procedimiento puede hacer que al finalizar el procedimiento aún queden pedidos sin servir. En este caso, se intenta insertar los pedidos pendientes, tanto la posición de su nodo pickup como delivery, en la posición que genere la mayor utilidad a la ruta.

De esta forma se generan 7 columnas iniciales por cada vehículo, es decir, 7*k* columnas ingresan inicialmente al PMr.

A continuación, se presenta en detalle la descripción de la heurística utilizada para resolver cada subproblema.

**3.2. Descripción de la heurística para *k*-SP**

La heurística desarrollada se divide en dos grandes etapas secuenciales: inicialización y local search. En la etapa de inicialización, se crea una ruta inicial factible para cada uno de los *k*-SP. Esta ruta inicial, corresponde a la base sobre la cual se aplicarán diferentes modificaciones en la etapa de local search. Las modificaciones a una ruta se realizan mediante la aplicación de distintos operadores, los que son elegidos basados en su probabilidad la cual se va actualizando en el tiempo. La fase de local search termina cuando se alcanza el número máximo Ω de iteraciones o cuando se hayan ejecutado todos los operadores sin haberse generado mejoras en la función objetivo.

Un esquema general de la heurística propuesta se presenta a continuación:

**Algoritmo 1.-** Esquema general de heurística propuesta

*#Construcción de la solución inicial factible*

MejorSolucion Inicializacion()

MejorFO = FO(MejorSolucion) #*Mejor función objetivo*

[Se inicializan todos los operadores con equiprobabilidad]

***#****Fase de búsqueda local*

**while** (No se cumple criterio de término) **do**

**[**Se selecciona un operador según probabilidad]

SolucionActual Operador(MejorSolucion);

[Se actualizan las probabilidades de todas las heurísticas]

**if** (FO(MejorSolucion) > MejorFO) **then**

MejorSolucion SolucionActual;

MejorFO = FO(SolucionActual);

**end**

**end**

A continuación, se describen en profundidad cada una de las etapas mencionadas anteriormente, los operadores utilizados en la fase de local search así como la manera en la cual se actualizan las probabilidades de cada uno.

**3.2.1 Inicialización**

En esta etapa, se escoge aleatoriamente una ruta inicial para cada *k-*SP desde el conjunto de todas las rutas factibles para el respectivo avión *k* ingresadas hasta ese momento al PMr (columnas del PMr) y se recalcula su utilidad en base a la función objetivo del *k-*SP (eq 23). De esta forma, a medida que avanzan las iteraciones, el conjunto de rutas candidatas a ser solución inicial aumenta en cantidad.

**3.2.2 Local search**

Para la fase de local search se realizan modificaciones a rutas específicas sin tomar en cuenta la existencia de las rutas creadas por otros aviones hasta ese momento. La interacción con los demás subproblemas se da a través de la resolución del PMr, el cual proporciona los valores duales para la función objetivo del subproblema.

En esta fase, se emplean siete diferentes operadores que modifican la ruta inicial (columna) ingresada. Para que una modificación en una ruta sea aceptada, ésta debe mejorar la utilidad de la ruta dada por la ecuación (23). De estos siete operadores, cuatro consisten en el intercambio de posición de pedidos dentro de una misma ruta, el quinto busca intercambiar un pedido de una ruta con otro sin asignar, mientras que los últimos dos operadores cambian el número de pedidos en la ruta ya sea mediante la inserción de un pedido nuevo o por la eliminación de un pedido existente. Esto será explicado con mayor detalle en la sección 3.2.2.1.

La elección de cada operador en la etapa de local search se realiza de manera aleatoria basada en la eficiencia ponderada de cada operador que se define como:



Donde es la eficiencia ponderada del operador *h* en la iteración anterior y corresponde a la eficiencia en la iteración actual del operador que se define como:



donde representa la mejora porcentual lograda en la función objetivo, producto del empleo del operador *h*, es el tiempo que toma en ejecutarse dicho operador y corresponde a un valor muy pequeño para evitar que un operador que no genera mejoras en la función objetivo en una iteración quede con nula probabilidad de ser elegida en la siguiente. El parámetro es definido por el modelador para dar mayor o menor peso a la eficiencia presente o histórica del operador. Finalmente, la probabilidad de elección de un operador *h* queda definida por:



De esta forma se le entrega una mayor probabilidad de elección a aquellos operadores que en el pasado han logrado una mayor eficiencia ponderada en la mejora de la función objetivo.

**3.2.2.1Operadores**

En esta sección se describen los siete operadores utilizados en la fase de local search. El detalle de cada uno de ellos se explica a continuación:

Intra-route operators

Los cuatro operadores de intercambio de pedidos corresponden a las intra-route operators utilizados en Cherkesly et al (2015): Intra-route request exchange, modifica las posiciones de dos pedidos *i* y *j* en una ruta específica intercambiando las posiciones de *pickup* *i+* y *j+* y las de *delivery* *i-* y *j-*. Intra-route request relocate, busca relocalizar un pedido *i* en su mejor posición dentro de la ruta. Las últimas dos heurísticas Intra-route multiple request exchange y Intra-route multiple request relocate corresponden a intercambio y reubicación de rutas truncadas o subrutas de cada pedido. Una ruta truncada se define como un subconjunto de una ruta que tiene como nodo inicial el *pickup* de un pedido *i* y como nodo final el *delivery* del pedido *i.* Además, cumple con la condición de que no exista otra carga sobre el avión al salir del nodo inicial, y que la carga al salir del nodo final es cero.

Swap:

Este operador, basado en el *swap routed and covered customers* de Talerian and Salari (2015), intercambia un pedido perteneciente a una ruta *k* con otro pedido que pertenece a la lista *Lp* de pedidos pendientes. Esta lista *Lp* corresponde a aquellos pedidos que no han sido incorporados a la ruta hasta ese momento, ordenados de mayor a menor ganancia unitaria. La ganancia unitaria para un pedido *i* está dada por , donde corresponde a la tarifa asociada al pedido *i* y al costo dual obtenido de la ecuación (20) en el PMr. Para una ruta *k* dada, este operador recorre la lista *Lp* buscando para cada pedido *i Lp* su mejor posibilidad de intercambio con otro pedido *j* perteneciente a la ruta *k*, siempre y cuando mejore la utilidad total de la ruta. Notar que este operador no modifica ni el tamaño de la ruta *k* ni el de la lista *Lp*, sino solamente los pedidos que pertenecen a cada una de ellas.

Drop\_and\_Add:

Este operador, como el propuesto por Talerian and Salari (2015), aumenta la longitud de la ruta, insertando pedidos sin asignar pertenecientes a *Lp* sobre una ruta *k*. Para cada pedido *i Lp*, este operador lo agrega en su mejor posición de inserción siempre y cuando mejore la utilidad total de la ruta.

Delete

Para una ruta *k*, se recorre la ruta y se van eliminando pedidos de manera de ver si la eliminación de ciertos pedidos mejora la utilidad total de la ruta.

Los operadores empleados en la etapa de local search generan modificación en la secuencia de nodos que se visitan, pero no optimizan la asignación de cargas. Como el modelo propuesto permite que los aviones lleven cargas parciales desde un nodo y éstos pueden ser visitados por más de un avión, se plantea el siguiente método para la asignación de carga:

AsignarCargas:

Cada operador descrito anteriormente genera una ruta *r* con su secuencia de nodos a los cuales es necesario asignar una cantidad de carga que puede ser recogida o entregada (), a modo de cumplir con las restricciones de carga (8)-(11) del problema original. Este método tiene una funcionalidad híbrida, debido a que en primer lugar asigna de manera greedy el total de la demanda a cada nodo de forma secuencial, de acuerdo al orden de visita en la ruta. En caso de superar la capacidad del avión se resuelve el siguiente *problema de optimización lineal* para efectuar una reasignación de la carga de todos los nodos de la ruta:



donde es el conjunto de pedidos que forman parte de la ruta *r.*

El objetivo de este problema (27) es maximizar las ganancias de transportar cada uno de los pedidos asociado a la ruta. La restricción (28) permite que no se asigne carga a los nodos inicial y final de la ruta, asociados al inicio y fin del horizonte de planificación. La restricción (29) asegura que la carga del nodo *pickup* y *delivery* sea la misma para cada pedido. Las restricciones (30) y (31) establecen que la carga asociada a un *pickup* sea no-negativa y menor a la demanda total de ese pedido. La continuidad de la carga está restringida mediante (32) y (33). La restricción a la capacidad del avión está dada por (34) y la naturaleza de las variables está dada por (35) y (36).

El procedimiento secuencial propuesto resulta ser más rápido que resolver el problema AC en los casos donde la restricción de capacidad no está activa.

**4. RESULTADOS COMPUTACIONALES**

La math heuristic propuesta fue programada en Python 2.7 y se utilizó Gurobi 6.5 para resolver los modelos de optimización (PMr) en cada iteración. El computador utilizado para correr las diferentes instancias está equipado con Intel Core i5 (2.6 GHz) y con 8 GB de Memoria RAM.

Para todos las instancias testeadas se usa un valor de Ω=35, , y se considera una flota fija de 7 aviones.

**4.1 Benchmark**

Para probar el desempeño de la math heuristic propuesta, se compararon los resultados de este procedimiento contra dos benchmark: i) MIP: solución del modelo de optimización (1)-(18) a través de MIP solver y ii) Fix route: solución del modelo (1)-(18) pero fijando las rutas de la planificación base. Este segundo benchmark pretende dimensionar el impacto de operar la misma planificación original después de las disrupciones, permitiendo únicamente optimizar la cantidad de carga a llevar de cada uno de los pedidos. Debido a que el modelo debe ser usado para fines operacionales donde las decisiones deben tomarse en cortos periodos de tiempo, se impuso un tiempo límite de solución de 2 horas. Ambos benchmark fueron resueltos con el solver Gurobi.

En el caso de la math heuristic, debido a la naturaleza aleatoria en la elección de columnas iniciales y de operadores en la etapa de búsqueda local, resulta muy probable que ejecuciones sucesivas de la misma instancia arrojen resultados diferentes. Además, para un periodo de tiempo determinado (por ejemplo, una hora) es posible, en ciertos casos, que la math heuristic sea ejecutada más de una vez. Tomando en cuenta lo anterior, se impuso como criterio de término para la math heuristic el primero en ocurrir entre: i) : máximo número de corridas o ejecuciones globales, ii) Δ: número de iteraciones globales sin mejora en la función objetivo y iii) τ: tiempo máximo de ejecución. De esta forma se fijaron .

**4.1 Instancias**

Para testear nuestro modelo se modificaron las instancias AA propuestas por Ropke and Cordeau (2009) para el PDPTW. Este grupo está compuesto por 10 instancias en total con el número de pedidos que varía entre los 30 y 75 pedidos. Todas las instancias consideran un horizonte de planificación T=600, una longitud de la ventana de tiempo W=60 y capacidad idéntica para todos los vehículos de CAP=15. Para cada avión se establece su depot inicial y final de manera aleatoria. Las coordenadas (x,y) se determinan de acuerdo a una distribución U[0,50]. De esta manera, para elaborar la planificación base se considera la condición inicial de pedidos y ubicación de los aviones al inicio y término del horizonte de planificación. Las instancias con disrupciones se construyen a partir de modificaciones que sufre la condición inicial que pueden ser de dos tipos: i) modificación de la magnitud de demanda (peso) de pedidos originales y ii) aparición de nuevos pedidos. Los pedidos originales ven modificada su demanda original de acuerdo a un factor. Si , el pedido no sufre modificación, si modifica su demanda y si el pedido desaparece (su demanda es nula luego de la disrupción). En el caso que el pedido modifique su demanda, la nueva demanda viene dada por con .

Para cada instancia se generan nuevos pedidos con y <1. Para cada nuevo pedido se genera su posición tanto de *pickup* como de *delivery* de acuerdo a una distribución uniforme en el cuadrado [0,50]x[0,50] y su demanda . Para definir las ventanas de tiempo para un pedido *i,* se comienza fijando el earliest time () para el nodo *pickup* de acuerdo a una distribución uniforme en el intervalo [0,T-], siendo el tiempo de viaje directo entre su nodo *pickup* y *delivery*, y luego se calcula su lastest time () como . Luego las ventanas de tiempo para el nodo *delivery* se generan de acuerdo a y .

Debido a que en el caso con disrupción, la aerolínea tiene la opción de no servir o servir parcialmente algunos pedidos si estos no son rentables, se debe generar para cada pedido una tarifa unitaria. Para generar la tarifa de cada pedido supondremos que el costo total por llevar un pedido completo *i* desde su nodo *pickup* al nodo *delivery* () representa un porcentaje θ de la tarifa total *Fi* ( con . Luego, la tarifa unitaria *fi* vendrá dada por .

Para este trabajo, los valores de los diferentes parámetros se fijaron en (0.3,0.85,0.7,1.2,0.2,0.05,0.2).

Notar que aún cuando ciertos pedidos pueden desaparecer (demanda nula), estos deben ser considerados de igual manera en el caso con disrupción, ya que pueden constituir parte de la planificación base. Esto puede ocurrir en situaciones en que la tripulación debe terminar su función, tienen como lugar de descanso un aeropuerto específico o el avión tiene que ser sometido a algún tipo de mantenimiento.

En la tabla 1 se presenta un resumen de las 10 instancias indicando para cada una de ellas, el número de pedidos originales, que se generan, desaparecen y modifican su peso.

Tabla 1: Características de las instancias analizadas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **originales** | **nuevos** | **desaparecen** | **modifican peso** |
| AA30 | 30 | 5 | 4 | 12 |
| AA35 | 35 | 6 | 2 | 18 |
| AA40 | 40 | 5 | 8 | 13 |
| AA45 | 45 | 2 | 9 | 20 |
| AA50 | 50 | 2 | 7 | 24 |
| AA55 | 55 | 3 | 7 | 26 |
| AA60 | 60 | 9 | 8 | 25 |
| AA65 | 65 | 11 | 10 | 26 |
| AA70 | 70 | 3 | 9 | 36 |
| AA75 | 75 | 3 | 17 | 30 |

Para cada instancia se testean tres posibles valores de penalidad (=20,15,10), para representar casos en que se permite mayor o menor flexibilidad en los cambios a la planificación base.

**4.3 Análisis de Resultados**

En la Tabla 2 se muestran los resultados para las instancias analizadas. La primera columna presenta el nombre de la instancia indicando el tipo de instancia (AA), número de pedidos y el valor de la penalidad considerada (S=20,15,10). Las columnas dos a cuatro muestran los resultados para el primer benchmark MIP, las columnas cinco a ocho los resultados de la math heuristic propuesta y las columnas nueve a once los resultados del segundo benchmark FIX route. Para las tres comparaciones se reporta el valor de la función objetivo, el tiempo de solución y el gap respecto al upper bound encontrado por el MIP durante las dos horas de ejecución. Notar que este upper bound puede no necesariamente corresponder a la mejor solución entera al problema. Para la math heuristic también se reporta el número de iteraciones globales efectuadas. Para cada instancia, se destaca en negrita la mejor solución encontrada entre las tres comparaciones.

Tabla 2: Resultados comparativos para las diez instancias analizadas considerando diferentes valores de penalidad.



De la Tabla 2 se observa que en 24 de las 30 instancias analizadas la math heuristic entregó el mayor valor de la función objetivo. De estos, en dos de ellos logra un resultado idéntico al MIP e igual a la solución óptima del problema (AA30\_S20 y AA45\_S20). De las seis instancias restantes, en cuatro de ellas el MIP arroja los mejores resultados en término de función objetivo, que coincide con instancias que tienen pocos pedidos (entre 30 y 40) y penalidad alta. En estas instancias la math-heuristic se encuentra en promedio a 0.63% de la solución del MIP, nunca excediendo este porcentaje en más de 1%. Por su parte, el FIX route es únicamente superior en dos instancias (AA70\_S15 y AA70\_S10). Tomando en cuenta todas las instancias, la math heuristic obtiene ganancias en promedio 33% mayores que las obtenidas por el MIP y 9.53% mayores que FIX route.

A su vez, el gap promedio con respecto al upper bound de la math-heuristic, considerando todas las instancias, alcanza un 5.89% que es considerablemente menor a los obtenidos tanto por el MIP (31.15%) como por el FIX route (14.96%). Lo anterior muestra que en general el enfoque propuesto arroja mejores resultados que los benchmark analizados.

Comparando los resultados para distintas magnitudes de la penalidad, se observa que en general los resultados del MIP empeoran a medida que disminuye el valor de la penalidad. Incluso el benchmark FIX route arroja valores de la función objetivo más altos que el MIP para instancias mayores a 35 pedidos en el caso de S=15 y con más de 30 pedidos para S=10. Una razón que podría explicar estos resultados radica en que, al ser la penalidad menor, el problema se vuelve menos restrictivo, producto de que habrían más alternativas posibles de ruteo, y por tanto más difícil de resolver. Por esto es posible que durante el periodo de ejecución no se encuentre una solución suficientemente buena. Es más, hay instancias como AA45\_S15 y AA65\_15 donde el MIP no alcanza a encontrar una solución factible entera en el tiempo de impuesto de solución.

Respecto a los tiempos computacionales, la math-heuristic propuesta alcanza a realizar varias iteraciones globales para instancias pequeñas de hasta 45 pedidos. Para instancias mayores, el número de iteraciones globales empieza a disminuir logrando hacer una sola en las instancias con mayor número de pedidos. En la tabla se pueden observar tiempos que sobrepasan el límite impuesto de 3600 segundos. Esto se debe a la condición de término especificada, la cual se verifica antes de entrar a una nueva iteración global. Luego si el tiempo es menor a 3600 segundos e ingresa a la iteración global, ésta se efectúa completamente y se vuelve a verificar la condición en la siguiente iteración. Lo anterior explica por qué hay instancias (por ejemplo, AA45\_S15) que efectúan más de una iteración global y además muestran tiempos sobre el límite impuesto. Respecto a los benchmark, los resultados muestran que en el caso del MIP, solo cuatro instancias logran ser resueltas a optimalidad en el tiempo propuesto mientras que el benchmark FIX route obtiene los menores tiempos al corresponder a una versión restringida del problema original donde únicamente se rutea la carga en una red con vuelos fijos.

A partir de los resultados presentados anteriormente se puede observar que la math heuristic propuesta entrega mejores resultados que los dos benchmark propuestos en los tiempos computacionales impuesto. Lo anterior sugiere que el modelo planteado puede ser usado como una herramienta operacional que ayude a la toma decisiones.

**6. CONCLUSIONES**

En este paper abordamos el problema de ajustes de último minuto a los itinerarios de aviones cargueros cuando ocurren disrupciones en la demanda de corto plazo. Para esto se formuló un modelo reactivo con fines operacionales basado en el Pickup and Delivery with Time Windows (PDPTW) que resuelve simultáneamente las etapas de Diseño de Itinerarios, Ruteo de Aviones y Ruteo de Carga, a la vez que considera las tripulaciones mediante penalidades por cambiar ciertos tramos de vuelo de la planificación base.

Propusimos y desarrollamos una math-heuristic para resolver el problema, mediante el uso simultáneo de técnicas de modelos de programación matemática y heurísticas. En particular, se desarrolló un método de Generación de Columnas, donde los subproblemas se resuelven de manera heurística, para determinar la ruta óptima de cada avión.

Para evaluar el desempeño de la metodología propuesta generamos diez instancias con entre 30 y 75 pedidos las cuales se evaluaron considerando tres niveles distintos de penalización por no respetar la planificación base, generando un total de 30 instancias en total. Comparamos la math heuristic propuesta contra dos benchmark MIP y FIX route imponiendo un tiempo límite de dos horas. A partir de los resultados se concluye que la math heuristic propuesta obtiene mejores resultados en 24 de las 30 instancias analizadas, con ganancias en promedio 33% mayores que las obtenidas por el MIP y 9.53% mejores que FIX route tomando en cuenta todas las instancias analizadas. En aquellas instancias en donde no logra la mejor solución, el gap respecto a la mejor solución encontrada por los benchmark no supera el 1%. Lo anterior sugiere que el uso combinado de técnicas de programación matemática y heurísticas puede generar soluciones de muy buena calidad en tiempos computacionales acotados.

Como futura línea de investigación se plantea estudiar un enfoque de planificación robusta de itinerarios cargueros que considere los posibles cambios en la demanda al momento de su construcción.

**6. REFERENCIAS BIBLIOGRÁFICAS**

Barnhart, C., Johnson, E., Nemhauser, G., Salvelsbergh, M. y Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3), 316-329.

Barnhart, C., Belobaba, P., y Odoni, A. (2003). Applications of operations research in the air transport industry. *Transportation Science*, 37(4), 307–324.

Berge, M. y Hopperstad, C. (1993). Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research,* 41(1). 153-168.

Boeing (2014). World Air Cargo Forecast 2014-2015.

Bratu, S. y Barnhart, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9(3), 279–298.

Cherkesly, M., Desaulniers, G. y Laporte, G. (2015). A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. *Computers and Operation Research*, 62, 23–35.

Cordeau, J.-F., Desaulniers, G., Desrosiers J., Solomon M.M. y Soumis F. (2002). VRP with time windows. En: Toth P, Vigo D (eds.), *The Vehicle Routing Problem, SIAM monographs on discrete mathematics and applications* (pp. 157-193). Filadelfia: SIAM.

Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3), 573–586.

Dantzig, G. y Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research,* 8(1), 101-111.

Derigs, U., Friederichs, S. y Schäfer, S. (2009). A new approach for air cargo network planning. *Transportation Science*, 43(3), 370–380.

Derigs, U. y Friederichs, S. (2013). Air cargo scheduling: integrated models and solution procedures. *OR Spectrum*, 35(2), 325–362.

Desrosiers, J. y Lübbecke, M.E. (2005). *Column Generation* (1ra. ed.). New York: Springer.

Dumas, Y., Desrosiers, J. y Soumis, F. (1991). The pickup and delivery problem with time windows, *European Journal of Operational Research*, 54, 7–22.

Etschmaier, M. y Mathaisel, D. (1985). Airline scheduling: an overview. *Transportation Science*, *19*(2), 127–138.

Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR***,** 8(4), 407–424.

Froyland, G., Maher, S. y Wu, C.L. (2014). The recoverable robust tail assignment problem. *Transportation Science,* 48(3), 351–372.

Gao, C., Johnson, E., y Smith B. (2009). Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1), 2–16.

IATA (2015a). Cargo eChartbook Q3.

IATA (2015b). Air Passenger Market Analysis, December 2015.

Labadie, N., Prins, C., & Prodhon, C. (2016). *Metaheuristics for Vehicle Routing Problems*. John Wiley & Sons.

Lan, S., Clarke, J.-P. y Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1), 15–28.

Levin, A. (1971). Scheduling and fleet routing models for transportation systems. *Transportation Science*, 5(3), 232–255.

Lin, C. y Chen, Y. (2003). The integration of Taiwanese and Chinese air networks for direct air cargo services. *Transportation Research Part A: Policy and Practice*, 37(7), 629–647.

Lohatepanont, M. y Barnhart, C. (2004). Airline schedule planning: integrated models and algorithms for schedule design and fleet assignment. *Transportation Research*, 38(1), 19-32.

Maniezzo, V., Stützle, T., Voß, S. eds. (2010) Matheuristics – Hybridizing Metaheuristics and Mathematical Programming. Annals of Information Systems, vol. 10. Springer

Marsten, R. y Muller, M. (1980). A mixed-integer programming approach to air cargo fleet planning. *Management Science*, 26(11), 1096–1107.

Parragh, S. N., Doerner, K. F. y Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, 58(1), 21–51

Pita, J., Barnhart, C. y Antunes, A. (2013). Integrated flight scheduling and fleet assignment under airport congestion. *Transportation Science*, 47(4), 477–492.

Qu, Y. y Bard, J. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers and Operations Research*, 39(10), 2439–2456.

Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. y Krishnamurthy, N. (2000). Airline fleet assignment with time windows. *Transportation Science*, 34(1), 1–20.

Ropke, S. y Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, *40*(4), 455–472.

Ropke, S., Cordeau, J.-F. y Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4), 258–272.

Ropke, S. y Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3), 267–286

Rosenberger, J., Johnson, E. y Nemhauser, G. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, 37(4), 408–421.

Rosenberger, J., Johnson, E. y Nemhauser, G. (2004). A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3), 357–368.

Savelsbergh, M. y Sol, M. (1995). General pickup and delivery problem. *Transportation Science*, 29(1), 17.

Sigurd, M., Pisinger, D. y Sig M. (2004). Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38, 197–209.

Talebian, M. y Salari, M. (2015). A GRASP algorithm for a humanitarian relief transportation problem. *Engineering Applications of Artificial Intelligence*, 41, 259–269.

Tang, C.-H., Yan, S. y Chen, Y.-H. (2008). An integrated model and solution algorithms for passenger, cargo, and combi flight scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 44(6), 1004–1024.

Wada, M., Delgado, F. and Pagnoncelli, B. (2017). A risk averse approach to the capacity allocation problem in the airline cargo industry. *Journal of the Operational Research Society*, 68(6), 643-651

Wolsey, L. A. (1998). *Integer Programming* (1ra. ed.). New York: Wiley.

Xu, H., Chen, Z.-L., Rajagopal, S. y Arunapuram, S. (2003). Solving a practical pickup and delivery problem. *Transportation Science*, 37(3), 347–364.

Yan, S., Chen, S. y Chen, C.-H. (2006). Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E: Logistics and Transportation Review*, 42(5), 409–430.

Yan, S. y Chen, C.-H. (2008). Optimal flight scheduling models for cargo airlines under alliances. *Journal of Scheduling*, 11(3), 175–186.