

**MODELO DE OPTIMIZACIÓN PARA LA PROGRAMACIÓN DE RUTAS Y VEHÍCULOS DE UNA
EMPRESA DE TRANSPORTE DE PACIENTES PUERTA A PUERTA EN LA CIUDAD DE
BOGOTÁ**

**CHRISTIAN DANILO CASTRO ORJUELA
JUAN NICOLÁS ROMERO BOBADILLA**

Director
ANGÉLICA SARMIENTO LEPESQUEUR
Ingeniera Industrial, M. Sc

**ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
INGENIERÍA INDUSTRIAL
BOGOTÁ D.C.
2020**

CONTENIDO

LISTA DE ILUSTRACIONES	2
LISTA DE TABLAS.....	2
1. INTRODUCCIÓN.....	3
2. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN	3
2.1. Preguntas de investigación	3
2.2. Objetivos.....	3
3. ESTADO DEL ARTE	4
4. DEFINICIÓN DEL PROBLEMA Y FORMULACIÓN MATEMÁTICA.....	5
4.1. Características y supuestos del problema	5
4.2. Formulación matemática	6
5. RESULTADOS	9
6. ANÁLISIS DE RESULTADOS.....	12
7. CONCLUSIONES.....	13
8. INVESTIGACIONES FUTURAS	13
9. BIBLIOGRAFÍA.....	14
ANEXOS.....	15

LISTA DE ILUSTRACIONES

Ilustración 1. Instancia inicial (10 servicios y 4 vehículos).....	10
Ilustración 2. Solución de la instancia inicial (10 servicios y 4 vehículos). .	11
Ilustración 3. Tiempos de generación (GAMS).....	12
Ilustración 4. Tiempos necesarios para obtener las soluciones óptimas (GAMS).....	12

LISTA DE TABLAS

Tabla 1. Parámetros de las instancias generadas (U = Urbanos, C = Camionetas, G = V. Grandes, R = V. con Rampa, NU = No Urbanos, RES = V. con Restricción).....	10
---	----

1. INTRODUCCIÓN

La programación de rutas y vehículos para el transporte de pacientes es un problema que engloba factores logísticos como el ruteo de vehículos y la búsqueda de reducir tiempos, costos o distancias recorridas; con factores propios del sector salud como el cumplimiento y el adecuado trato a los pacientes.

En cuanto al ruteo de vehículos, el problema presenta dos características principales: primero, la inclusión de rangos de tiempo para la recogida de los pacientes, lo cual se conoce como Time Windows (Ventanas de Tiempo), en el cual, se tiene un intervalo de tiempo en el cual se debe llegar a cada destino. Las ventanas de tiempo pueden ser de dos tipos:

- Soft time windows: en este tipo de ventanas de tiempo, el servicio se puede realizar fuera del límite de las ventanas de tiempo, considerando penalidades en dichos casos.
- Hard time windows: en este tipo de ventanas de tiempo, no se pueden realizar los servicios fuera del intervalo establecido por las ventanas de tiempo.

Segundo, debido a que cada usuario se debe recoger y llevar a las localizaciones requeridas sin realizar paradas intermedias, podría asimilarse a un Problema de Recogida y Entrega (PDP), el cual, como su nombre indica define un problema en el cual se debe construir un conjunto de rutas para satisfacer la recogida y entrega mediante el uso de una flota determinada.

Debido a la naturaleza de los problemas PDP, el caso que se presenta es una variación del Problema de Recogida y Entrega con Ventanas de Tiempo (PDPTW), el cual es un problema NP-Hard y, por lo tanto, encontrar la solución óptima para instancias de gran volumen de datos, en un tiempo de ejecución razonable resulta imposible. Por lo cual, para los problemas NP-Hard se recomienda el uso de algoritmos heurísticos que permitan la obtención de resultados de alta calidad, en tiempos de ejecución mucho más cortos.

2. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

2.1. Preguntas de investigación

- ¿Cómo se puede caracterizar, por medio de un modelo matemático, el conjunto de rutas que componen la operación diaria de una empresa dedicada al transporte de pacientes desde su casa a un centro médico y viceversa?
- ¿Hasta qué punto es viable la utilización de un método exacto de optimización para obtener una solución de un problema de este tipo?

2.2. Objetivos

2.2.1. Objetivo general

Diseñar un modelo de optimización para el transporte de pacientes puerta a puerta (casa – centro médico y viceversa) y analizar la conveniencia de la utilización de un método exacto para su solución dependiendo del tamaño del problema.

2.2.2. Objetivos específicos

- Formular un modelo de programación lineal entera-mixta para la programación de rutas y asignación de vehículos que permitan satisfacer la demanda de servicios de transporte de pacientes, considerando las implicaciones de capacidad, ventanas de tiempo, requerimientos de los usuarios y restricciones de movilidad.
- Solucionar con un enfoque exacto, varias instancias aleatorias del modelo, modificando el número de servicios y con ello el tamaño del problema.
- Determinar, mediante la comparación de los tiempos de ejecución, el punto en el que deja de ser viable la utilización de métodos exactos para encontrar la solución óptima de un problema NP-Hard como éste.

3. ESTADO DEL ARTE

En esta sección se presenta la revisión de la literatura académica relacionada con el Problema de Recogida y Entrega con Ventanas de Tiempo (PDPTW), así como sus aplicaciones al transporte de pacientes puerta a puerta, los métodos empleados en su solución y los tiempos de procesamiento relacionados.

El PDPTW surge como una variante del Problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW), cuyas restricciones principales están relacionadas con la capacidad y disponibilidad de los vehículos, las relaciones de precedencia entre los puntos de origen y destino, la necesidad de partir de un depósito y volver a este mismo, los requerimientos de asignar determinado tipo de vehículo a un servicio en el caso de las flotas heterogéneas y, por supuesto, las ventanas de tiempo [1].

En [2] se presenta la descripción y la formulación general del Problema de Recogida y Entrega (PDP), el cual, consiste en la construcción de una serie de rutas que permitan satisfacer un conjunto de solicitudes de transporte, o servicios, con los puntos de origen y destino de la carga definidos, así como su magnitud. En este caso, la carga debe ser transportada por un vehículo sin ningún tipo de transbordo. Vale la pena mencionar que los problemas PDP suponen una mayor complejidad que los problemas VRP, debido a que en estos últimos los transportes se realizan siempre partiendo del depósito, haciendo que los servicios que tengan un destino cercano sean asignados a un mismo vehículo de una forma más sencilla. De igual forma, en ese artículo se muestra una revisión de los métodos de solución para cada una de las variantes del PDP.

Más específicamente, el problema del transporte puerta a puerta de pacientes o personas discapacitadas, como aplicación del PDPTW, ha sido solucionado por distintos autores mediante la utilización de varios métodos heurísticos. En [3], se trata un problema basado en la necesidad de atender n servicios, teniendo en cuenta que existen tres tipos de personas discapacitadas a transportar (sin silla de ruedas, con silla de ruedas plegable y con silla de ruedas no plegable), ventanas de tiempo y varios tipos de vehículos con diferentes capacidades para cada tipo de persona. Este es resuelto mediante el agrupamiento de los servicios en *mini-clusters*, definidos como la serie de servicios atendidos desde que el vehículo que los atiende deja de estar vacío hasta vuelve a estarlo, de forma que se asegure que el tiempo necesario para recorrer las distancias dentro y entre ellos sea mínima. El método propuesto en este artículo solucionó un problema con 2545 servicios, considerados como los existentes en un día promedio de operación en Toronto, con un tiempo de procesamiento de 5 horas.

De forma similar, Paolo Toth y Daniele Vigo definen en [4] el Problema de Transporte de Personas Discapacitadas (HTP) como una generalización del Problema de Recogida y Entrega con Ventanas de Tiempo (PDPTW). En este artículo, se soluciona un problema similar al presentado en [3], con la inclusión de algunos aspectos como la necesidad de algunas personas de transportarse con personal de acompañamiento y la posibilidad de trasladar usuarios en taxi como última opción. De hecho, el objetivo en este caso es minimizar el uso de estos vehículos junto con los costos totales de la operación. El método de solución propuesto consiste en el uso de un algoritmo heurístico de inserción paralela, mediante el cual primero se elige una serie de servicios pivote, a los cuales se les añaden progresivamente los demás, para obtener así la solución final del problema; además, se utiliza un algoritmo *Tabu Thresholding* con el fin de mejorar los valores objetivo obtenidos. Con la utilización de este procedimiento se logró solucionar problemas con cerca de 300 servicios en cuestión de segundos.

Más recientemente, en [5] se investiga un problema de transporte en el sistema de salud pública de Hong Kong muy similar al descrito en [4], con la adición de restricciones de disponibilidad del personal de asistencia y los conductores, incluyendo aspectos como turnos de trabajo y descansos de almuerzo. En este caso, el problema es solucionado mediante un algoritmo metaheurístico de búsqueda local iterativa y vecindad variable (ILS-VND), el cual consiste en la generación de una solución inicial y el mejoramiento sucesivo de esta hasta alcanzar un objetivo deseado. Este método consiguió solucionar 249 instancias de forma sucesiva, para un total de 41.993 servicios en un tiempo de poco más de una hora y media.

En general, las publicaciones relacionadas con la solución de problemas de transporte de pacientes o personas discapacitadas coinciden en que estos son una aplicación del Problema de Recogida y Entrega con Ventanas de Tiempo (PDPTW) y están encaminados al diseño de algoritmos heurísticos y metaheurísticos que permitan conseguir soluciones cada vez más cercanas a las óptimas y tiempos de procesamiento menores que den lugar a la solución de problemas más grandes.

4. DEFINICIÓN DEL PROBLEMA Y FORMULACIÓN MATEMÁTICA

Para la definición de las características del problema se tiene como base la operación diaria de una empresa dedicada al transporte de pacientes, en trayectos que van desde su vivienda hasta algún centro de atención médica y viceversa en la ciudad de Bogotá y sus alrededores.

Basados en las características generales del PDPTW mencionado anteriormente, y teniendo en cuenta las consideraciones específicas del problema de transporte en cuestión, se presentan a continuación las características y los supuestos en los que se basa la formulación del modelo matemático, que tiene como objetivo la minimización de la distancia total recorrida por los vehículos estando vacíos.

4.1. Características y supuestos del problema

- La programación de las rutas, o servicios, y la asignación de vehículos se realiza a diario para todos los recorridos del día siguiente.
- Se conocen de antemano las ubicaciones de origen y destino de los servicios, así como la ventana de tiempo en la que es conveniente recoger al paciente al inicio de cada uno.
- Cada vehículo parte de un depósito diferente y debe volver a él al finalizar su programación de servicios del día.

- Existen tres tipos de vehículos: camionetas, vehículos grandes o con espacio para sillas de ruedas plegadas y vehículos con rampa. Dependiendo de las necesidades de cada paciente, se deberá asignar un vehículo de cierto tipo a la ruta.
- Las rutas urbanas tienen como característica la restricción de movilidad para algunos vehículos, independientemente de su tipo.
- Existe un máximo de tiempo diario que un vehículo puede estar en operación, el mismo para todos los días de la semana.
- Un vehículo no puede atender más de un servicio al mismo tiempo.
- Existen recorridos de ida y vuelta, o redondos; sin embargo, ambos trayectos se consideran por separado y pueden ser realizados por distintos vehículos.
- Todos los servicios del día deben ser atendidos. Los costos no son relevantes.

4.2. Formulación matemática

Conjuntos:

I = Conjunto de rutas o servicios $\{1, 2, \dots, n\}$

El número total de servicios varía dependiendo de la instancia del modelo a analizar.

J alias de **I**

K = Conjunto de vehículos $\{1, 2, \dots, m\}$

El número total de vehículos varía dependiendo de la instancia del modelo a analizar.

G = Conjunto de tipos de vehículo {Camioneta, Grande, Con rampa}

Variables de decisión:

$w_k = 1$ si el vehículo k es utilizado

$s_{k,i} = 1$ si el vehículo k utiliza el arco desde su depósito hacia el origen del servicio i

$p_{k,i,j} = 1$ si el vehículo k utiliza el arco desde el destino del servicio i hacia el origen del servicio j

$o_{k,j} = 1$ si el vehículo k utiliza el arco desde el destino del servicio j hacia su depósito

$x_{i,k} = 1$ si el servicio i es atendido por el vehículo k

$u_i =$ Variable de nivel o actor medible del servicio i

$h_i =$ Instante de tiempo en el que se inicia el recorrido del servicio i

Parámetros determinísticos:

$A =$ Tiempo máximo que cualquier vehículo puede permanecer en operación

$C_{i,j} =$ Distancia de recorrido entre el destino del servicio i y origen del servicio j

$T_i =$ Distancia del servicio i

$E_i =$ Tiempo de apertura de la ventana de tiempo para iniciar el recorrido del servicio i

$L_i =$ Tiempo de cierre de la ventana de tiempo para iniciar el recorrido del servicio i

$C1_{k,i} =$ Distancia entre el depósito del vehículo k y el punto de origen del servicio i

$C2_{k,j} =$ Distancia entre el punto de destino del servicio j y el depósito del vehículo k

$V_{k,g} = 1$ si el vehículo k es de tipo g

$R_{i,g} = 1$ si el servicio i debe ser atendido por un vehículo de tipo g

$B_i = 1$ si el servicio i es un recorrido urbano

$D_k = 1$ si el vehículo k no tiene restricción de movilidad

Función objetivo:

$$\text{Min } Z = \sum_k \sum_i \sum_j C_{i,j} p_{k,i,j} + \sum_k \sum_i C1_{k,i} s_{k,i} + \sum_k \sum_j C2_{k,j} o_{k,j} \quad (1)$$

La función objetivo (1) del modelo de programación entera mixta consiste en minimizar el tiempo total recorrido en vacío, correspondiente a los trayectos entre rutas, así como aquellos desde y hasta los depósitos de cada vehículo al iniciar y terminar la jornada, respectivamente.

Restricciones:

$$\sum_k \sum_j p_{k,j,i} + \sum_k s_{k,i} = 1; \forall_i \quad (2)$$

$$\sum_k \sum_j p_{k,i,j} + \sum_k o_{k,j} = 1; \forall_i \quad (3)$$

$$\sum_i p_{k,i,j} + s_{k,i} = \sum_i p_{k,i,j} + o_{k,j}; \forall_j, \forall_k \quad (4)$$

$$\sum_j p_{k,i,j} + s_{k,i} = x_{i,j}; \forall_k, \forall_i \quad (5)$$

$$\sum_i x_{i,k} \leq w_k * 1000; \forall_k \quad (6)$$

$$\sum_i x_{i,k} \geq w_k; \forall_k \quad (7)$$

$$\sum_i s_{k,i} = w_k; \forall_k \quad (8)$$

$$\sum_j o_{k,j} = w_k; \forall_k \quad (9)$$

$$\sum_i s_{k,i} = \sum_j o_{k,j}; \forall_k \quad (10)$$

$$u_i - u_j + n * p_{k,i,j} \leq n - 1; \forall_i, \forall_j, \forall_k \quad (11)$$

El primer grupo de restricciones garantiza que el ruteo se programe correctamente. Las ecuaciones (2) y (3) aseguran que algún vehículo llega al origen y sale del destino de cada ruta, garantizando además que todos los servicios son atendidos; a su vez, la expresión (4)

asegura que es el mismo vehículo el que realiza estos dos movimientos, el cual según la restricción (5) es el que atiende el servicio asociado a esa ruta.

Las expresiones (6), (7), (8), (9) y (10) garantizan que los vehículos que se utilizan, es decir, aquellos que se asignan a por lo menos un servicio, salen de su depósito al iniciar su recorrido y regresan a este al finalizarlo.

La restricción (11) evita que se generen rutas cíclicas particionadas, o subrutas, garantizando que se genere solo un recorrido para cada vehículo, el cual comienza con la salida del depósito y termina con el regreso a este mismo, atendiendo los servicios asignados y realizando los trayectos entre las rutas de estos en el medio.

$$h_i \leq L_i; \forall_i \quad (12)$$

$$h_i \geq E_i; \forall_j \quad (13)$$

$$h_i + T_i + C_{i,j} \leq H_j + 1000 * (1 - \sum_k p_{k,i,j}); \forall_i, \forall_j \quad (14)$$

El siguiente conjunto de restricciones corresponde a los requisitos de tiempo de los servicios. Las ecuaciones (12) y (13) garantizan que la recogida de cada paciente se da dentro de la ventana de tiempo de su servicio; mientras que la restricción (14) asegura que el vehículo que lo va a atender está disponible y tiene tiempo suficiente para llegar al punto de origen de la ruta.

$$\sum_i s_{k,i} * C1_{k,i} + \sum_i x_{i,k} * T_i + \sum_i \sum_j p_{k,i,j} * C_{i,j} + \sum_j o_{k,j} * C2_{k,j} \leq A; \forall_k \quad (15)$$

$$\sum_{k \notin V_{k,g}} x_{i,k} = 0; \forall_g, \forall_i \in R_{i,g} \quad (16)$$

$$\sum_{i \in B_i} x_{i,k} = 0; \forall_k \notin D_k \quad (17)$$

El tercer grupo de restricciones consiste en las condiciones propias del contexto del problema. La restricción (15) garantiza que los vehículos utilizados cumplen con el límite de tiempo máximo que pueden estar en operación por día. La ecuación (16) asegura que los servicios son asignados a algún vehículo de cierto tipo de acuerdo con las necesidades de cada paciente. Además. La expresión (17) evita que los servicios urbanos sean realizados por vehículos con restricción de movilidad en la ciudad.

$$w_k \in \{1,0\}; \forall_i \quad (18)$$

$$s_{k,i} \in \{1,0\}; \forall_k, \forall_i \quad (19)$$

$$p_{k,i,j} \in \{1,0\}; \forall_k, \forall_i, \forall_j \quad (20)$$

$$o_{k,j} \in \{1,0\}; \forall_k, \forall_j \quad (21)$$

$$x_{i,k} \in \{1,0\}; \forall_i, \forall_k \quad (22)$$

$$u_i \geq 0; \forall_i \quad (23)$$

$$h_i \geq 0; \forall_i \quad (24)$$

Finalmente, el último conjunto de restricciones que establecen el tipo de cada variable. En los casos de las ecuaciones (18), (19), (20), (21) y (22), corresponden a las variables binarias, y las expresiones (23) y (24), a las positivas.

5. RESULTADOS

Como se explicó anteriormente, el modelo corresponde a un problema NP-Hard, cuya complejidad y tiempo de encontrar una solución óptima aumentan a medida que se tiene una mayor cantidad de servicios y vehículos a programar. Por esta razón, se generaron distintas instancias del problema, aumentando en 10 unidades la cantidad de servicios en cada una de ellas, con el fin de determinar el tiempo de generación y el tiempo total necesario para encontrar la solución óptima en cada una de ellas.

Para la generación de estas instancias se tuvieron en cuenta los siguientes supuestos:

- Los puntos de origen de los servicios, así como la ubicación de los depósitos, se generan de forma aleatoria uniforme en coordenadas cartesianas que van desde (0,0) km hasta (120,120) km.
- Los puntos de destino de los servicios se generan mediante vectores cartesianos que comienzan en el punto de origen de cada servicio y terminan en el punto de destino. Los componentes rectangulares pueden tener un valor entre -30 km y 30 km, generados de forma aleatoria uniforme.
- Los tiempos de transporte son proporcionales a la distancia tipo Manhattan entre los puntos de origen y destino, con una velocidad promedio de 60 km/h (1 km/min).
- Las ventanas de tiempo para el inicio de un servicio tienen una duración de 60 minutos.
- La línea de tiempo para el inicio de las ventanas de tiempo se define entre las 5:00 a.m. y las 7:00 p.m. y se maneja en minutos, donde las 5:00 a.m. corresponden a 0 y las 7:00 p.m., a 840. Los tiempos de inicio para cada servicio son generados de forma uniforme a lo largo de esta línea de tiempo.
- Se establece una relación de 1 vehículo por cada 3,8 servicios.
- 86% de los servicios requieren vehículos tipo camioneta, 10% requieren vehículos grandes y 4%, vehículos con rampa.
- Los vehículos disponibles se distribuyen de la siguiente forma según su tipo: 67% son camionetas, 27% son vehículos grandes y 6% son vehículos con rampa.
- 20% de los vehículos tienen restricción de movilidad zonas urbanas. Con el fin de generar equidad entre las diferentes instancias, estos vehículos corresponden solamente a camionetas, seleccionadas de forma aleatoria.
- Los servicios urbanos con restricción de movilidad representan un 95% del total. Los servicios correspondientes al 5% restante son seleccionados de forma aleatoria entre los que requieren camionetas.
- Teniendo en cuenta que en las primeras instancias algunos aspectos toman un valor menor a 1 debido a los porcentajes mencionados anteriormente, se asegura que exista al menos un servicio que requiera vehículo con rampa, así como la disponibilidad de al menos uno de estos últimos.

En la Ilustración 1 se presenta a modo de ejemplo la primera instancia generada, correspondiente a 10 servicios y 4 vehículos, con las flechas de color negro representando las rutas de los distintos servicios. Así mismo, se muestran en la Tabla 1 los parámetros utilizados en la generación de cada una de las instancias.

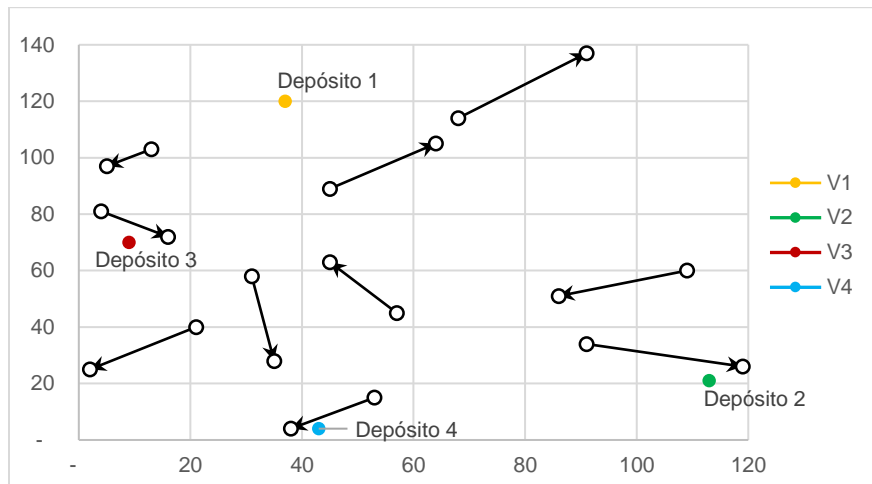


Ilustración 1. Instancia inicial (10 servicios y 4 vehículos). Fuente: Elaboración Propia.

INSTANCIAS										
SERVICIOS	U	C	G	R	NU	VEH	C	G	R	RES
10	10	8	1	1	0	4	2	1	1	0
20	19	17	1	1	1	6	4	1	1	1
30	29	25	3	1	1	8	5	2	1	1
40	38	33	4	1	2	11	7	3	1	1
50	48	41	5	2	2	13	9	3	1	2
60	57	49	6	2	3	16	11	4	1	2
70	67	58	7	2	3	18	12	5	1	2
80	76	65	8	3	4	21	14	6	1	3
90	86	74	9	3	4	24	16	6	2	3
100	95	82	10	3	5	26	17	7	2	3

Tabla 1. Parámetros de las instancias generadas (U = Urbanos, C = Camionetas, G = V. Grandes, R = V. con Rampa, NU = No Urbanos, RES = V. con Restricción). Fuente: Elaboración Propia.

Las instancias generadas fueron solucionadas en un computador personal con procesador AMD A10-9600P de 2.40 GHz con memoria RAM de 16 GB y sistema operativo de 64 bits, utilizando el Sistema General de Modelaje Algebraico (GAMS) versión 24.1.

De esta forma, se obtuvo la solución óptima para cada una de las instancias, consiguiendo así la programación de los vehículos con el fin de atender todos los servicios generados en cada caso. En la Ilustración 2 se muestra como ejemplo la solución de la instancia inicial, en la que se puede

apreciar la serie de desplazamientos que debe realizar cada vehículo para atender los servicios que le fueron asignados.

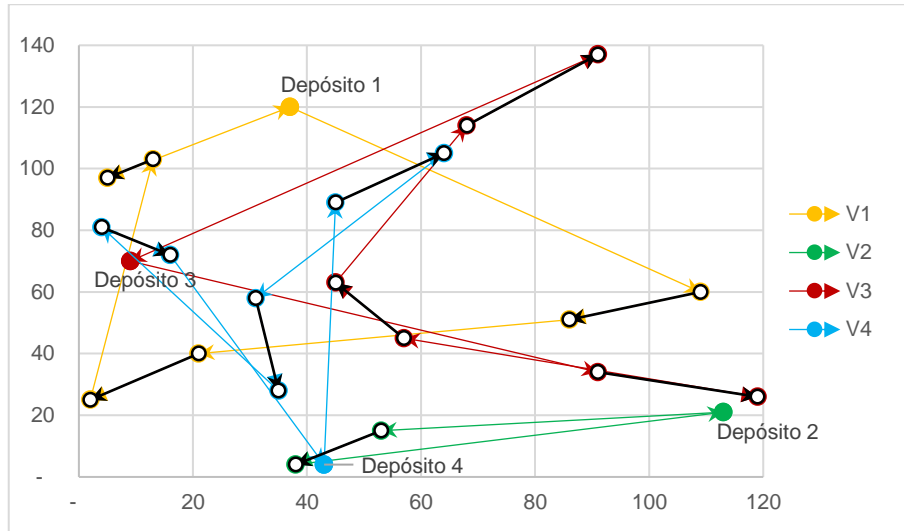


Ilustración 2. Solución de la instancia inicial (10 servicios y 4 vehículos). Fuente: Elaboración Propia.

Cabe destacar que la distancia que se deseaba minimizar es la correspondiente a las líneas de color que conectan los recorridos de los servicios mostrados en la Ilustración 1 y los depósitos. Aunque a primera vista los arcos de la solución no parezcan ser los óptimos, se debe recordar que en el caso de este problema los servicios están separados no solo en espacio sino también en el tiempo.

Así mismo, se registró también el tiempo de generación del problema y el tiempo total necesario para encontrar la solución óptima, ambos mostrados por GAMS, para cada una de las instancias. Los resultados de estos datos se muestran en su totalidad en el Anexo 1 y resumidos en las Ilustraciones 3 y 4.

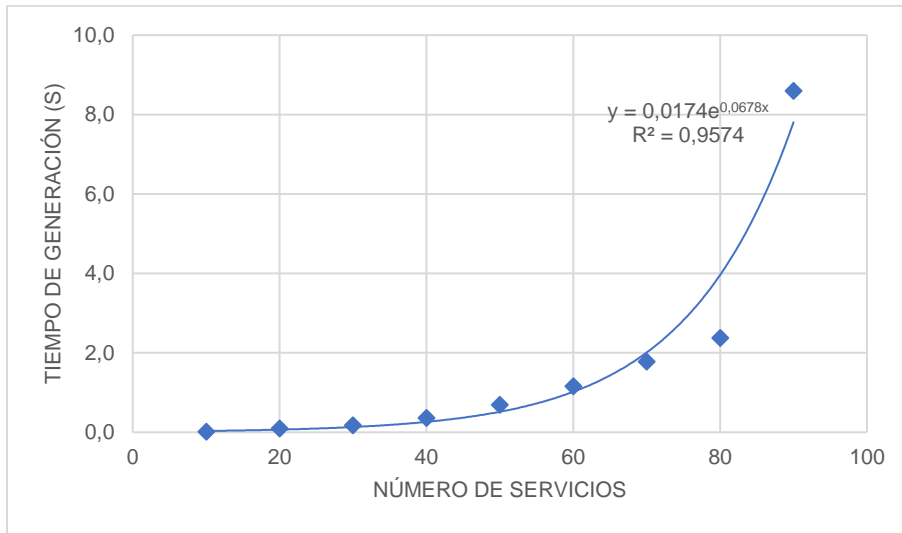


Ilustración 3. Tiempos de generación (GAMS). Fuente: Elaboración Propia.

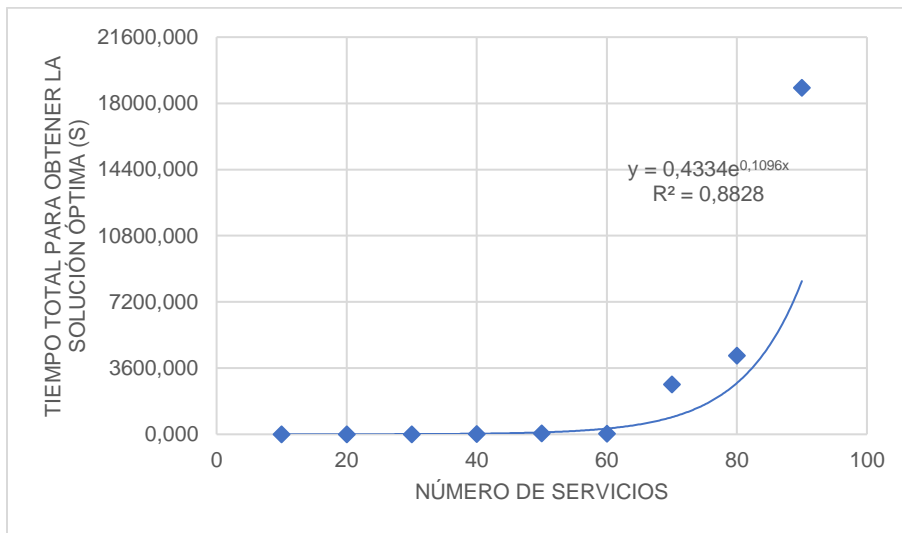


Ilustración 4. Tiempos necesarios para obtener las soluciones óptimas (GAMS). Fuente: Elaboración Propia.

Los tiempos correspondientes a la última instancia generada (100 servicios y 26 vehículos) no se muestran puesto que se obtuvo un error pasadas casi las cinco horas de procesamiento (4h:57m:55s), debido a que el solver CPLEX superó el límite de memoria permitido por causa del tamaño del problema.

6. ANÁLISIS DE RESULTADOS

En la Ilustración 3 se puede observar que los tiempos de generación de las primeras 8 instancias son crecientes, pero no presentan mayor variación entre uno y otro. Es solo hasta la novena instancia que se presenta la variación característica del comportamiento exponencial de los problemas NP-Hard.

Por otra parte, en la Ilustración 4 se puede observar cómo el tiempo necesario para solucionar el problema aumenta de forma similar a un aumento exponencial a medida que el número de servicios a realizar durante el día aumenta; concordando con la característica principal de un NP-Hard, la cual indica que este tipo de problemas no se pueden solucionar en tiempos polinómicos.

También, a diferencia de los tiempos de generación, es apreciable que el aumento del tiempo necesario para encontrar la solución óptima comienza a ser notable desde la séptima instancia, pasando de los 39 segundos de la instancia anterior a los 2720 (45,3 minutos) de esta última.

Además, posteriormente se logra identificar que desde los 90 servicios por día los tiempos de ejecución aumentan a más de 5 horas, y que incluso al llegar a los 100 se vuelve imposible la solución óptima del problema, probando así que el uso de un modelo exacto es poco práctico para la planeación de la operación diaria de la empresa si se tiene en cuenta que este proceso debe ser realizado a diario.

7. CONCLUSIONES

- El uso de métodos exactos para problemas NP-Hard es útil para observar el comportamiento del modelo y para solucionar instancias pequeñas del problema; sin embargo, a la hora de procesar gran volumen de datos son poco prácticos debido al tiempo requerido para encontrar la solución óptima del problema.
- Se logró obtener la solución óptima para un problema de transporte de pacientes puerta a puerta de hasta 90 servicios utilizando un método exacto en algo más de cinco horas. Mediante este mismo método no fue posible encontrar la solución óptima para una instancia de 100 servicios.
- La implementación de un método heurístico podría resultar útil debido a que permitiría procesar mayor cantidad de servicios en un tiempo razonablemente menor, permitiendo a la empresa realizar su programación diaria por este medio.
- Fue posible identificar que la aleatoriedad juega un papel importante en el tiempo necesario para solucionar el problema. Es posible que se genere un conjunto de nodos que resulte más complejo de resolver que otro, tal como sucedería en la vida real, en la que no se puede controlar la distribución de estos debido a que dependen de los requerimientos de cada paciente.
- La duración de las ventanas de tiempo también incide directamente en la dificultad y el tiempo de solución del problema. Si son muy ajustadas, aumenta considerablemente la complejidad e incluso puede ocurrir que no exista ninguna solución factible para el problema al reducir su duración en unos cuantos minutos.
- Durante la generación de las instancias se evidenció que existe una posibilidad mucho mayor de que la solución de este problema no sea factible cuando se tiene una cantidad reducida de servicios a atender, debido a que también se tiene poca disponibilidad de vehículos y es muy posible que los servicios se crucen en un momento en el que todos están ocupados, igual que en la realidad.

8. INVESTIGACIONES FUTURAS

Teniendo en cuenta que en este caso se buscó siempre la solución óptima de las diferentes instancias del problema, se podrían llevar a cabo experimentos en los que se incluya una tolerancia

que permita también conseguir una solución aproximada, de forma que se analice el impacto en los tiempos de ejecución a medida que cambia esta tolerancia.

En la realidad, en algunos casos existe la posibilidad de que algunos pacientes acepten compartir vehículo con otro que se dirija al mismo centro médico o que, en todo caso, realice una ruta similar. Por lo tanto, se propone incluir este aspecto en el problema de transporte de pacientes puerta a puerta con el fin de determinar si este hace aumentar la complejidad del problema al ofrecer más posibilidades de solución o si por el contrario la hace disminuir.

9. BIBLIOGRAFÍA

- [1] M. KÜÇÜK y Ş. TOPALOĞLU YILDIZ, «A Constraint Programming Approach for the Pickup and Delivery Problem with Time Windows,» *Pamukkale University Journal of Engineering Sciences*, vol. 25, nº 9, pp. 1041-1049, 2019.
- [2] M. W. P. Savelsbergh y M. Sol, «The General Pickup and Delivery Problem,» *Transportation Science*, vol. 29, nº 1, pp. 17-29, 1995.
- [3] I. Ioachim, J. Desrosiers, Y. Dumas, M. M. Solomon y D. Villeneuve, «A Request Clustering Algorithm for Door-to-Door Handicapped Transportation,» *Transportation Science*, vol. 29, nº 1, pp. 63-78, 1995.
- [4] P. Toth y D. Vigo, «Heuristic Algorithms for the Handicapped Persons Transportation Problem,» *Transportation Science*, vol. 31, nº 1, pp. 60-71, 1997.
- [5] A. Lim, Z. Zhang y H. Quin, «Pickup and Delivery Service with Manpower Planning in Hong Kong Public Hospitals,» *Transportation Science*, vol. 51, nº 2, pp. 688-705, 2017.
- [6] C. Y. C. Wan Chen Chiang, «Considering the Performance Bonus Balance in the Vehicle Routing Problem with Soft Time Windows,» *Procedia Manufacturing*, vol. 11, pp. 2156-2163, 2017.

ANEXOS

INSTANCIAS												
SERVICIOS	U	C	G	R	UN	VEH	C	G	R	RES	TIEMPO DE GENERACIÓN (s)	TIEMPO PARA OBTENER LA SOLUCIÓN ÓPTIMA (s)
10	10	8	1	1	0	4	2	1	1	0	0,016	4,984
20	19	17	1	1	1	6	4	1	1	1	0,093	5,603
30	29	25	3	1	1	8	5	2	1	1	0,172	5,897
40	38	33	4	1	2	11	7	3	1	1	0,359	21,138
50	48	41	5	2	2	13	9	3	1	2	0,687	47,154
60	57	49	6	2	3	16	11	4	1	2	1,156	39,249
70	67	58	7	2	3	18	12	5	1	2	1,781	2720,003
80	76	65	8	3	4	21	14	6	1	3	2,375	4285,506
90	86	74	9	3	4	24	16	6	2	3	8,594	18844,879
100	95	82	10	3	5	26	17	7	2	3	6,266	NSE

Anexo 1. Tiempo de generación y tiempo necesario para obtener la solución óptima de las instancias (NSE = No Se Encontró Solución).