

Optimización de la distribución de hielo de una empresa Hidalguense en el Estado de México

Product distribution optimization of an Hidalgo's small ice factory in the State of Mexico

PÉREZ-RAMÍREZ, Carlos Manuel†* & RUIZ-ESCAMILLA, Pedro César

Tecnológico Nacional de México/Instituto Tecnológico de Pachuca

ID 1^{er} Autor: *Carlos Manuel, Pérez-Ramírez* / ORC ID: 0000-0002-3544-9073, Researcher ID Thomson: P-2410-2018, CVU CONACYT ID: 931752

ID 1^{er} Coautor: *Pedro César, Ruiz-Escamilla* / ORC ID: 0000-0002-8759-0124, Researcher ID Thomson: P-1986-2018, arXiv Autor ID: clucker92

Recibido: Septiembre 27, 2018; Aceptado: Diciembre 06, 2018

Resumen

Para minimizar la distancia total que recorre un vehículo repartidor de bolsas de hielo a un máximo de 40 clientes de una amplia zona geográfica, se ha construido un modelo de Programación Lineal Entera Binaria en una aplicación de hoja de cálculo cuyo optimizador determina el orden de visita a cada uno. El problema se modela como un gráfico dirigido en el que los nodos son los clientes y los arcos son la red de carreteras que los une: como el nodo de partida y terminación de la ruta es la planta en la que se fabrica y embolsa el hielo, y cada nodo se visita exactamente una vez, la estructura corresponde al denominado Problema del Agente Viajero Asimétrico. Al ser un total de cuarenta clientes se ha elaborado un modelo de 41 nodos que se utiliza como plantilla para resolver la red de aproximadamente veinte sitios a visitar en cada ocasión. El ahorro esperado en la distancia recorrida por viaje es del 9%, estimado de una muestra de recorridos basados en la experiencia del chofer. En un segundo modelo se obtiene la ruta de mínimo tiempo de recorrido, con un ahorro del 13%.

Optimización, Transporte, Agente via-jero

Abstract

To minimize the total distance traveled by a delivery vehicle of ice bags to a maximum of 40 customers in a wide geographical area, a binary integer linear programming model has been built in a spreadsheet application whose optimizer determines the order to visit each client. The problem is modeled as a directed graph in which the nodes are the customers and the arcs are the road network that connects them: as the starting and ending node of the route is the plant where the ice is manufactured and bagged, and each node is visited exactly once, the structure corresponds to the so-called asymmetric traveler salesman problem. With a total of forty customers, a model of forty-one nodes has been developed and used as a template to solve the network of approximately twenty customers to visit each time. The expected savings in distance traveled on a given route is around 9%, estimated from a sample of tours based on the driver's experience. In a second model the route of minimum travel time is obtained, with a similar savings (13%).

Optimization, Transportation, Traveling-salesman

Citación: PÉREZ-RAMÍREZ, Carlos Manuel & RUIZ-ESCAMILLA, Pedro César. Optimización de la distribución de hielo de una empresa hidalguense en el Estado de México. Revista de Tecnologías en Procesos Industriales. 2018, 2-5: 24-36

*Correspondencia al Autor (Correo Electrónico: cperez@itpachuca.edu.mx)

† Investigador contribuyendo como primer autor.

Introducción

Una de las causas de ineficiencias administrativas y operativas de las micro, pequeñas y medianas empresas mexicanas es el bajo o nulo nivel de utilización de procedimientos, métodos y técnicas desarrollados por la ciencia y cuya eficiencia ya ha sido comprobada en la práctica.

La Programación Lineal (PL) es una de las ramas del campo científico conocido como Investigación de Operaciones y que se encarga de construir modelos matemáticos para diversas clases de problemas de optimización, así como de desarrollar y programar en computadora los algoritmos que los resuelvan.

En este artículo se reporta la implantación exitosa en hoja de cálculo de un modelo de PL para encontrar la ruta de mínima distancia (o, alternativamente, de mínimo tiempo de recorrido) para el reparto de bolsas de hielo que un día a la semana realiza la empresa Mr. Hielo a clientes del Estado de México (EdoMex) por vía terrestre en un camión, implicando ahorro en el gasto de combustible, contaminación, mantenimiento del vehículo y tiempo de recorrido, principalmente; además, haría posible aumentar el número de clientes a atender cada vez. Mr. Hielo tiene en total cuarenta clientes en el EdoMex, pero solamente un subconjunto de ellos –no siempre el mismo– es el que cada semana le requiere el producto y, generalmente, le es imposible satisfacer a todos ellos en un día de trabajo. Actualmente la empresa no puede dedicar otro día a estos clientes por razones de atención a sus demás compradores y también económicas (por uno, dos o tres clientes a los que no haya podido entregar la mercancía no le conviene enviar el camión al siguiente u otro día de la misma semana).

El modelo corresponde a la clase conocida como *Problema del Agente Viajero* (TSP: *Traveling Salesman Problem*), que está clasificado como un problema de solución computacional compleja (NP-difícil: el tiempo de solución computacional crece exponencialmente con el incremento del número de sitios de la ruta), y es por ello que además de la PL se han desarrollado y propuesto varios métodos de solución, tales como los algoritmos genéticos y las redes de neuronas, entre otros.

Sin embargo, cuando el tamaño del problema sea tal que un modelo de PL lo pueda resolver en un tiempo computacional aceptable (pocos segundos o minutos, por ejemplo), su uso tiene la ventaja de que se tendrá la mejor de todas las soluciones posibles (solución óptima), lo que no garantizan los demás enfoques mencionados (que son algoritmos heurísticos).

El artículo consta de nueve secciones: la primera es esta introducción; en la segunda se describen las características generales de la empresa para la que se realizó el trabajo y de la situación estudiada; luego, en la tercera, se mencionan los fundamentos del modelo desarrollado; en la cuarta se especifican las etapas para la implantación del modelo de PL en la aplicación informática de hoja electrónica de cálculo Calc del *software* libre OpenOffice; en la quinta sección se consignan los resultados obtenidos; y después las dedicadas a los agradecimientos, conclusiones, recomendaciones pertinentes a la implantación y mejoramiento del modelo y las referencias.

La empresa y situación de estudio

La fábrica de hielo Mr. Hielo fue fundada en 1990 bajo la razón social Distribuidora de Hielo Texcoco; posteriormente, en 1992, se estableció como Hielo San Antonio S.A. de C.V., reconocida simplemente como Mr. Hielo, fabricante y comercializadora de hielo, cuyo logo es el que se muestra en la figura 1. Actualmente vende hielo en diversas presentaciones a empresas de distintos giros comerciales en los estados de Hidalgo, México y Puebla, así como al público en general. Entre las varias presentaciones del producto se cuentan: barra purificada de 50 kg, barra de enfriamiento de 50 kg, barra de ósmosis de 50 kg, bolsa de rolito y cubo de 15 kg y bolsa de rolito de 5 kg.



Figura 1 Logotipo de Mr. Hielo

Fuente: Mr. Hielo

Mr. Hielo cuenta con oficinas administrativas y una sola planta de fabricación, ubicadas en Calle 24 de Octubre #6 Col. San Antonio El Desmonte, Pachuca, Hgo., México (figura 2).

En total cuenta con una plantilla de 53 trabajadores: 19 en producción con un nivel académico medio de secundaria; 20 en ventas con educación media de bachillerato; y 14 en administración, todos con título de licenciatura.



Figura 2 Ubicación de Hielo San Antonio S.A. de C.V.
Fuente: Google Maps

Para el reparto de hielo a sus clientes dispone de un solo vehículo sin frigorífico con capacidad de carga de 6 ton, y solamente los días jueves de cada semana hace la distribución a los del EdoMex. Aunque en la zona colindante de este estado con Hidalgo tiene una cartera de cuarenta clientes, en cada viaje surte a un número menor (la mitad o un poco más) por varias razones: no todos requieren resurtido del producto, capacidad de carga del camión, conservación del producto y el tiempo disponible. El estudio de esta ruta de reparto se prefirió sobre las demás por ser la de la zona con mayor concentración de clientes, de población y de tráfico vehicular (figura 3).

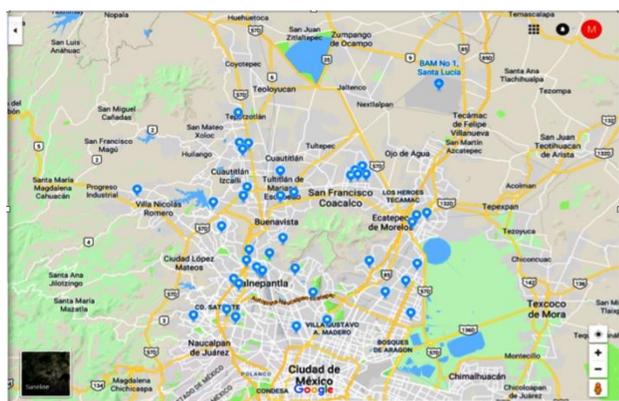


Figura 3 Zona de reparto (clientes indicados por los globitos azules)

Fuente: tecnología de Google Maps

ISSN: 2523-6822

ECORFAN® Todos los derechos reservados

Al inicio de la jornada de los jueves el área de ventas le da al chofer del vehículo – único tripulante y quien también se encarga de entregar las bolsas de hielo– la lista de clientes a suministrar la mercancía (en promedio 40 bolsas de 5 kg de rolitos de hielo a cada uno).

La secuencia en que se visitan los destinatarios la determina el mismo repartidor, basado en la experiencia de 5 años que ya posee sobre el mismo trabajo. Ocurre que en ocasiones, por las condiciones de tráfico, meteorológicas o de bolsas rotas por el manejo en las maniobras de descarga, no se alcanza a satisfacer la lista completa de entregas (20 destinos, en promedio) del día, con el problema adicional de que a su regreso a la planta el hielo no entregado ya se ha derretido.

Así, cada mañana de jueves el camión parte de la fábrica de hielo hacia el primer cliente, se detiene hasta realizar la entrega del lote, enseguida inicia el recorrido hacia el segundo cliente, etc. y finalmente regresa, por la noche del mismo día, a la fábrica. Teóricamente el problema tiene la estructura típica del modelo del agente viajero (Cook W. J., 2012): una situación muy conocida aunque complicada de resolver.

Fundamentos

El problema de los siete puentes de Königsberg (actual Kaliningrado, Rusia), fue un acertijo popular en el S. XVIII con interés científico-matemático, con el que Euler demostró, en 1736, la imposibilidad de una caminata que pasara una y solo una vez por cada puente. La figura 4 contiene un esquema del río Pregel con las dos islas, los siete puentes y su correspondiente grafo. Se reconoce el tratamiento euleriano de este problema como el comienzo de la *Teoría de Grafos* (Wikipedia, 2018).

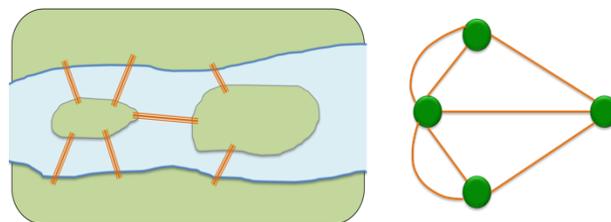


Figura 4 Puentes de Königsberg (izq) y grafo (der)

Un grafo es una estructura matemática que consta de un conjunto no vacío de una clase de objetos llamados *nodos* o *vértices* (círculos en la figura 4), una segunda clase de objetos denominados *arcos* o *aristas* (líneas en la misma figura), y una relación entre ambas clases. Un recorrido por todos los arcos, sin repetir ninguno es un *Camino Euleriano* y en caso de que sea cerrado (que termine en el nodo de inicio) se nombra *Circuito (o Ciclo) Euleriano*. El recorrido por los arcos *a, b, c, d, e, f, g, h* en el grafo del gráfico 1 constituye un *Circuito Euleriano*.

En el grafo de la figura 4 las aristas son los puentes y los vértices las porciones de tierra firme. Por lo tanto, la solución al acertijo sería cualquier *Circuito Euleriano* que se encontrara, lo cual resulta imposible.

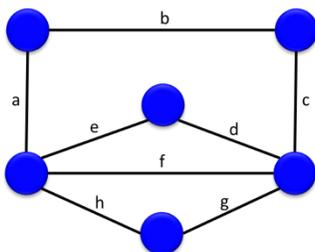


Gráfico 1 Circuito Euleriano

En cambio, un recorrido en el que se visiten todos los nodos, sin repetir ninguno, es un *camino Hamiltoniano* y si es cerrado se trata de un *circuito Hamiltoniano*. En el grafo del gráfico 2, la secuencia de nodos 1, 2, 3, 4, 5, 1 ilustra un circuito Hamiltoniano.

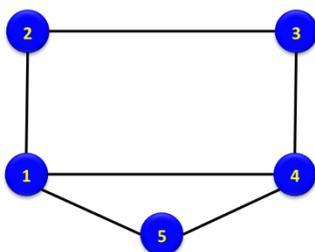


Gráfico 2 Circuito Hamiltoniano

El *problema del agente viajero*, TSP, es un caso muy estudiado por su enorme importancia práctica: un viajero (persona, vehículo, camión, autobús, yate turístico, mensajero, o cualquier otro objeto), debe visitar exactamente una sola vez cada lugar (poblado, sitio, casa, etc.) de una lista finita de ellos y regresar al punto de partida, al mínimo costo (monetario, distancia, tiempo) posible.

En un grafo los lugares a visitar son representados por los nodos y los arcos son las conexiones entre ellos, por lo que el objetivo es encontrar el circuito Hamiltoniano de mínimo costo. William J. Cook (2012) da un extenso panorama del TSP: los primeros intentos de vendedores, abogados y predicadores por resolverlo; los desarrollos formales gráficos de Leonhard Euler y del irlandés William R. Hamilton; pasando por el problema de 49 ciudades resuelto por Dantzig *et al* (Dantzig, Fulkerson, & Johnson, 1954) y los métodos exactos de Programación Lineal (entera, planos de corte, ramificación y acotamiento) y heurísticos; la creación, con tres colegas, del programa de cómputo *Concorde* para resolver TSP's simétricos de gran escala (85,900 ciudades); y la complejidad computacional del problema.

Un indicador de qué tan eficiente es un algoritmo para resolver un problema, es el orden de complejidad computacional, denotado por la llamada *función O grande* debido a su símbolo $O(\)$. En los años que siguieron a la invención del *método Simplex* para resolver modelos lineales continuos (LP's), proliferaron las propuestas de algoritmos adicionales para resolver modelos lineales enteros (IP's, desde problemas con costos fijos hasta los del agente viajero), que funcionaron muy bien para problemas relativamente pequeños, pero en general computacionalmente ineficientes a medida que aumentó el número de restricciones y de variables enteras. Entre estas propuestas se cuentan los planos de corte de Gómory, los algoritmos de programación dinámica, ramificación y acotamiento y otros apoyados en la teoría de grupos (Craig A., 2002).

Los algoritmos preferidos son los que se ejecutan en un tiempo máximo polinomial, mientras que los que finalizan en un tiempo finito no polinomial son clasificados como NP (*Non-deterministic Polynomial time*). Por ejemplo, un algoritmo evidente para encontrar el valor máximo de la lista 1, 2, 3, ..., n corre en un tiempo proporcional a n, por lo que su complejidad computacional es $O(n)$. El tiempo que requieren los algoritmos más rápidos para ordenar, creciente o decrecientemente, una base de datos de n registros de personas (nombre, domicilio, teléfono, etc.) por alguno de sus campos es proporcional a $O(n \log n)$.

Aunque el método Simplex de programación lineal teóricamente corre en tiempo exponencial $O(k^m)$, en la práctica ha resultado muy eficiente: el tiempo de ejecución es proporcional a $O(m^3)$, en donde m es el número de restricciones.

El gráfico 3 es un esquema de la clasificación de problemas con base en su complejidad computacional sobre la cual se advierte que, hasta ahora, los problemas de programación lineal entera (IP) son clasificados como difíciles de resolver (NP-Difícil, o NP-*Hard*) para los que no existe un algoritmo general eficiente. El problema del agente viajero es un IP (Craig A., 2002), y por lo tanto NP-Difícil, aunque muchos investigadores creen que es NP.

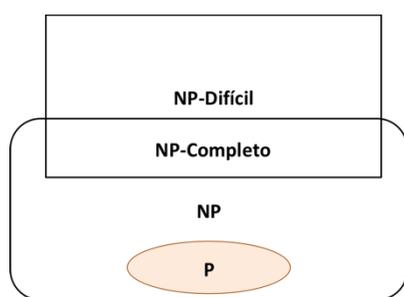


Gráfico 3 Clasificación de problemas

Si se llegara a demostrar que $P = NP$, entonces solo existirá la clasificación binaria P y NP-Difícil. A propósito, hay un premio de un millón de dólares al primero que resuelva en tiempo polinomial el TSP, lo que demostraría que $P=NP$ (Clay Mathematics Institute, 2007). Stephen Cook (2006) ofrece una descripción formal de la complejidad computacional P y NP .

El TSP ha sido intensivamente estudiado, porque aunque es fácil de enunciar y de verificar que un itinerario dado es factible, no es sencillo de resolver (encontrar el itinerario óptimo), aumentando la complejidad sus extensiones (restricciones adicionales de capacidad, tiempo, más de un viajero, distintos puntos de inicio o de finalización del recorrido, conexiones prohibidas, indeseables o inexistentes, etc.) lo que motiva el interés matemático y computacional. También porque hay móvil práctico: v. gr. diariamente, el programa de optimización meta-heurístico ORION, de UPS, especifica una ruta optimizada para cada uno de los 55,000 vehículos encargados de recoger y entregar los paquetes de ese día (Holland, Levis, Nuggehalli, Santilli, & Winters, 2017).

Hay aplicaciones artísticas, como la generación de retratos aproximados por puntos, reconocibles cuando son vistos a suficiente distancia (Chondl, 2012), o como el dibujo *Embrace* renderizado e impreso en acero inoxidable y latón de Robert Bosch (Bosch, 2010). Y hasta en la cinematografía: unos minutos antes de la hora y media de la película *Guardianes de la Galaxia*, el bandido azul, Yondu Udonta, utiliza su aparentemente inofensivo dardo para eliminar a un grupo de soldados alienígenas (incluido el que tripula un tanque flotante), mediante un letal recorrido por cada uno de los atacantes para finalmente regresar a las manos de Yondu (Chondl, 2016).

Dependiendo del tamaño y condiciones particulares del TSP, los métodos de solución pueden ser exactos o aproximados. Entre los primeros están la enumeración total de itinerarios posibles, la programación dinámica y la programación lineal entera y sus extensiones: relajamiento Lagrangiano, planos de corte, ramificación y acotamiento, corte y ramificación.

Entre los segundos, genéricamente denominados heurísticos o metaheurísticos, se cuentan el recocido simulado (*simulated annealing*), redes de neuronas (*neural networks*), algoritmos genéticos (*genetic algorithms*), búsqueda tabú (*tabu search*), colonia de hormigas (*ant colony algorithms*), búsqueda aleatoria (*random search*) y el enjambre de partículas (*particle swarm*). A veces se combinan: se recurre a una heurística para encontrar una buena solución que es utilizada como punto de partida de algún método exacto.

Cuando el costo de la conexión entre cada par de nodos del grafo es el mismo de ida y vuelta, el recorrido en el sentido de las manecillas del reloj es equivalente al de contrarreloj. En este caso se dice que el TSP es simétrico (STSP); de lo contrario el TSP es asimétrico (ATSP).

Por el método exacto de enumeración en un STSP de n nodos se tendría que valorar cada uno de los $(n-1)!/2$ posibles circuitos Hamiltonianos y $(n-1)!$ ciclos en el ASTP. En cualquier caso, el método se vuelve rápidamente ineficiente: en un ATSP habría que examinar solo 6 rutas con $n = 4$ nodos, pero si fueran $n = 11$ nodos serían 3,628,800 rutas.

Por la importancia práctica del TSP, un libro ampliamente utilizado en cursos de Investigación de Operaciones, (Taha, 2012), le dedica el capítulo once, ejemplificando algunos métodos, tanto exactos como heurísticos. Un segundo popular libro de texto (Hillier & Lieberman, 2010), de la misma disciplina, solo lo esboza en el tratamiento de los modelos de programación entera binaria en el undécimo capítulo.

En un documento de trabajo, Orman & Williams (2005) hacen un recuento de ocho distintas formulaciones de programación lineal entera del ATSP, entre las que destaca la del enfoque secuencial de Miller, Tucker y Zemlin, por dos razones: 1ª) se facilita su programación en hoja electrónica de cálculo; y 2ª) es suficiente para el tamaño del caso particular de Mr. Hielo. Enseguida se especifica dicho modelo.

Definiciones

$N = \{1, 2, 3, \dots, n\}$ Conjunto de nodos

$n =$ número de nodos a visitar

$x_{ij} = 1$ si el arco que va del nodo i al j es parte del itinerario (ruta)

$= 0$ si no lo es

$c_{ij} =$ costo de viajar del nodo i al j , $\forall i \neq j$

$= \infty$ si $i = j$

$u_i =$ variable continua: orden de visita al nodo i ($i \neq 1$)

Modelo

$$\text{Minimizar } \sum_{i,j \in N, i \neq j} c_{ij} x_{ij} \quad (1)$$

$$\sum_{j \in N, j \neq i} x_{ij} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{i \in N, i \neq j} x_{ij} = 1 \quad \forall j \in N \quad (3)$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad \forall i, j \in N - \{1\}, i \neq j \quad (4)$$

(1), (2) y (3) constituyen un modelo estándar de asignación, que es de la clase P; sin embargo, las restricciones (2) y (3) no garantizan que se obtendrá un ciclo hamiltoniano, sino que pueden dar lugar a varias subrutras (*subtours*), la cuales son evitadas por (4).

El modelo tiene $n(n-1)$ variables binarias 0-1, $(n-1)$ variables continuas y $n^2 - n + 2$ restricciones.

Metodología

Mr. Hielo, al igual que muchas MPyMES, presenta varias oportunidades de mejora, en todas sus áreas funcionales. De acuerdo con la alta administración se determinó analizar la distribución de bolsas de hielo de los jueves en el EdoMex por tener la mayor cantidad de clientes que las demás zonas de reparto en los estados de Hidalgo y Puebla.

Ya seleccionado el objeto de estudio, se procedió a establecer las fases apropiadas para el desarrollo del trabajo mostradas en la figura 5.

Planteamiento del problema. La situación de estudio es un problema de reparto de mercancías, en el que un solo vehículo inicia su recorrido desde la fábrica, entrega en secuencia los pedidos de varios clientes y regresa a la planta el mismo día.

El objetivo es reducir al mínimo la distancia del recorrido, en primera instancia, o el tiempo empleado en el mismo exceptuando el tiempo de entrega (porque es independiente del orden de visita a cada cliente) en segunda. Se trata de un problema muy conocido, fácil de verificar pero difícil de resolver: el del agente viajero.



Figura 5 Etapas del estudio

Investigación documental. De manera paralela al planteamiento del problema se inició la revisión de artículos, libros, *software* de optimización y de otras fuentes a fin de obtener información conceptual y teórica suficiente para seleccionar el modelo más apropiado a la situación de interés. En la sección fundamentos se sintetiza lo encontrado.

Selección del modelo de optimización. Debido a que el recorrido por carretera (tanto en distancia como en tiempo) desde un cliente A a uno B no es el mismo que de B a A ($c_{ij} \neq c_{ji}$, cuando $i \neq j$), y sin particularidades adicionales, es un TSP asimétrico (ATSP) para cuya solución se escogió el modelo de IP definido al final del apartado de la fundamentación teórica.

Datos. Como en cualquiera de las formulaciones de modelos revisadas se necesitan las distancias entre puntos del recorrido (o los tiempos), la fase de cálculo de las mismas se hizo a la par que la investigación sobre la elección del modelo específico a utilizar. Una forma sencilla, aunque algo laboriosa, de calcular distancias es mediante el posicionamiento geográfico por coordenadas de la planta y los clientes, pero la característica de asimetría descarta este procedimiento; por lo tanto se decidió recurrir a la tecnología de mapas de Google para esta labor, mucho más tardada que la del cálculo por coordenadas.

Selección del medio de programación. Aquí se trata de responder a la pregunta ¿en qué programar el modelo matemático seleccionado? Hay *software* comercial, tales como:

- AIMMS (<https://aimms.com>)
- AMPL (<https://ampl.com>)
- GAMS (<https://gams.com>)
- Gurobi (<http://www.gurobi.com>)
- LINGO (<https://www.lindo.com/>)

Entre varios más, que proveen un ambiente de modelado y al menos un optimizador, apropiados para resolver el modelo; pero se descartaron precisamente por tener que comprar una licencia útil (sus versiones de prueba son insuficientes para el tamaño del problema estudiado y además son para fines de enseñanza).

Las aplicaciones académicas tales como TORA, WinQSB y POM se dejaron fuera por su ineficiencia en el escalamiento (cambiar el tamaño) de modelos, además de estar diseñados para fines académicos.

En Internet hay optimizadores muy buenos *en línea* como los programas de acceso libre del NEOS Server, en <https://neos-server.org/neos/>, pero que requerirían un tiempo considerable de capacitación del personal (introducir datos, resolver e interpretar la solución), por lo que también se hizo a un lado. Ni siquiera se consideraron viables los sistemas informáticos matemáticos (Matlab, Mathematica, Maple, MathCad, etc.) por ser muy caros y se aprovecharía solo uno de sus módulos.

Las aplicaciones de hoja de cálculo MS Excel y OpenOffice Calc incluyen un optimizador llamado SOLVER, pero el de Excel es un *demo* que solo soporta hasta 200 variables por lo que es insuficiente en el caso bajo estudio (la versión estudiantil que acompaña a algunos libros de texto soporta hasta 500 variables, pero es para fines académicos, y para utilizar alguna versión de mayor capacidad habría que comprar la licencia).

También puede agregarse a Excel un optimizador gratuito denominado *OpenSolver* (<https://opensolver.org/>) pero en las pruebas preliminares resultó lento y con soluciones subóptimas.

El conjunto de programas de oficina *Apache OpenOffice*, cuya dirección (URL) es <https://www.openoffice.org/es/> es similar a *MS Office*, pero de código abierto y de distribución y uso libres (licencia Apache 2.0) para cualquier propósito (doméstico, negocios, educativa, administración pública) y multiplataforma (Mac, GNU/Linux y Windows), cuyo optimizador lineal Solver de la aplicación de hojas electrónicas de cálculo, llamada Calc, sí fue capaz de pasar las pruebas preliminares de tiempo de ejecución, exactitud, escalamiento y facilidad de uso, por lo que fue el medio elegido para programar y resolver los modelos ATSP.

Programación y resolución del modelo general. Para la programación en Calc, el modelo integrado por (1), (2), (3) y (4) en la sección de fundamentos se redefinió de manera conveniente.

Definiciones

$N = \{1, 2, 3, \dots, n\}$ Conjunto de fábrica y clientes (1 es la fábrica)

n = número de clientes a visitar, incluida la planta (41)

$x_{ij} = 1$ si al cliente j se le entrega el pedido inmediatamente después que al i

= 0 si no es así

c_{ij} = distancia, en km, de viajar por carretera del cliente i al j , $\forall i \neq j$

= M si $i = j$

u_i = variable continua: orden de visita al cliente i ($i \neq 1$)

Modelo

$$\text{Minimizar } \sum_{i,j \in N} c_{ij} x_{ij} \quad (5)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (6)$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N \quad (7)$$

$$u_i - u_j + 41x_{ij} \leq 40 \quad \forall i, j \in N - \{1\} \quad (8)$$

Este modelo adaptado tiene $41^2 = 1681$ variables de decisión 0-1, $2 \times 41 = 82$ restricciones de asignación y $40^2 = 1600$ restricciones de secuencia del recorrido.

Las distancias, en km, entre la planta y los cuarenta clientes, así como las que hay entre cada par de clientes se introdujeron en un primer arreglo tabular de 41 filas y 41 columnas de Calc, a excepción de la diagonal principal en cuyas casillas se escribió un número suficientemente grande comparado con las distancias entre clientes, M (v. gr.: $M = 500$), para evitar que un tramo del viaje inicie y termine en un mismo cliente (impedir *lazos* en términos teóricos).

En una segunda matriz del mismo tamaño que la primera se representaron las variables de decisión binarias x_{ij} del modelo. De este modo, la suma de los productos de los elementos correspondientes de ambas matrices rinde el escalar de la función objetivo (5). En una columna a la derecha de esta matriz se programó la suma de cada fila, y en la fila inmediata a la del último cliente la suma de cada columna, a fin de especificar las restricciones de asignación, (6) y (7).

Finalmente, en un tercer arreglo de la misma dimensión de los dos primeros, se programó el lado izquierdo de (8). La figura 6 es un esquema de esta distribución del modelo.

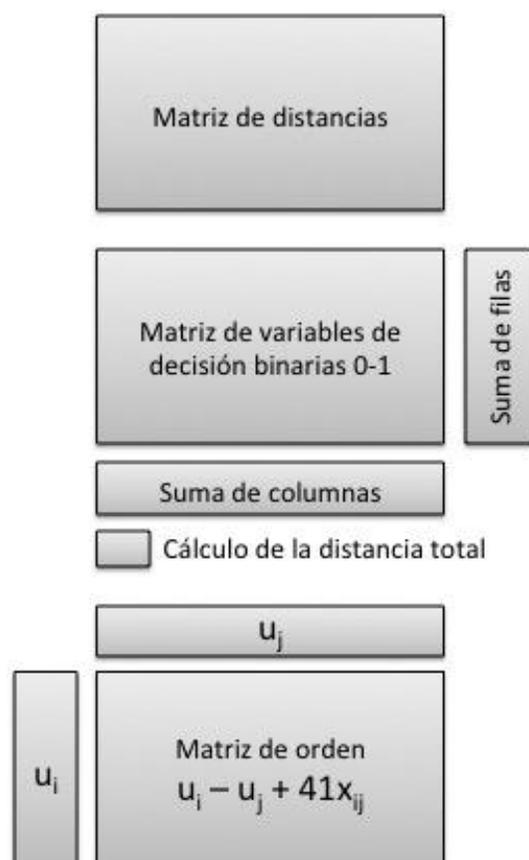


Figura 6 Esquema del modelo

En pseudocódigo el modelo de optimización es:

Min DistanciaTotal

SumaFilas = 1

SumaColumnas = 1

MatrizOrden ≤ 40 excepto fila 1 y columna 1

El problema de minimizar el tiempo de recorrido, exceptuando los tiempos de entrega, es más complicado que el de distancias, debido a que t_{ij} = tiempo, en minutos, que el camión invierte en recorrer el tramo carretero del cliente i al cliente j , es una variable aleatoria y ya no una constante. Con la tecnología de mapas de Google se obtuvieron tres tiempos con la finalidad de aproximar el comportamiento de esta variable: uno con tráfico fluido, otro con tráfico habitual y el tercero con tráfico lento.

En Calc, el modelo de tiempos es un duplicado del de distancias, reemplazando las distancias c_{ij} por los correspondientes tiempos de recorrido, en minutos, de un cliente a otro.

En este trabajo se consideraron los tiempos calculados con tráfico habitual.

Un análisis más completo tendría que combinar simulaciones de tiempos con patrones probabilísticos Triangulares o Betas con modelos de optimización a fin de diseñar un sistema dinámico de generación de rutas con tecnologías tipo GPS al inicio de cada viaje, y capaz de actualizar sobre la marcha la parte no realizada aún del recorrido.

Mr. Hielo identifica a sus clientes mediante una clave numérica, que se utilizó aquí tal cual. La tabla 1 es, recortada, la matriz de distancias de la figura 6. El orden en que aparecen las claves está relacionado con la cercanía de los clientes y se prefirió dejarlo así.

	C	D	E	F	G	H	AQ	AR
2		Datos: distancia, en kilómetros, desde un sitio enlistado en las filas hasta otro en las columnas						
3		Fábrica	775	722	784	924	136	139
4	Fábrica	500	76.4	75.1	70.2	70.2	79.2	75.4
5	775	76.2	500	5.1	3.8	7.8	13.7	9.9
6	722	83.2	6.3	500	7.8	8.9	19.4	15.6
7	784	72.4	4.5	6.3	500	3.7	16.4	12.6
8	924	70.3	8.7	8	4.7	500	20.7	16.9
43	1363	82.3	14.2	18.2	17.3	21.4	500	4.2
44	1392	78.3	10.2	14.1	12.4	15.9	3.8	500

Tabla 1 Distancias fábrica-clientes y entre clientes

La parte central del modelo de hoja de cálculo es la matriz de variables de asignación de los tramos de la ruta, y que en la figura 6 corresponde a los bloques centrales del esquema. La tabla 2 ilustra esta parte (no se muestra la mejor solución).

	C	D	E	F	G	H	AR	AS	AT	
48		Fábrica	775	722	784	924	1392		Suma	
49	Fábrica	0	0	0	0	0	0		1	
50	775	0	0	1	0	0	0		1	
51	722	0	0	0	1	0	0		1	
52	784	0	0	0	0	1	0		1	
53	924	0	0	0	0	0	0		1	
88	1363	0	0	0	0	0	1		1	
89	1392	0	0	0	0	0	0		1	
90										
91	Suma	1	1	1	1	1	1			
92										
93	Dist =	302.7 km								
94	Ruta:	Fábrica	1110	429	580	457	1580	994	Fábrica	
95			0	1	2	3	39	40		

Tabla 2 Variables 0-1 de asignación y ruta

La tabla 3 es la porción final: la matriz que se utiliza para asegurar que no se obtengan subrutas.

	A	B	C	D	E	F	G	AQ	AR
99				U1	U2	U3	U4	U4	U4
100		Orden visita		0	33	34	35	28	29
101				Fábrica				136	139
102	u1	0	Fábrica	0	-33	-34	-35	-28	-29
103	u2	33	775	33	0	40	-2	5	4
104	u3	34	722	34	1	0	40	6	5
105	u4	35	784	35	2	1	0	7	6
143	u40	28	1363	28	-5	-6	-7	0	40
144	u41	29	1392	29	-4	-5	-6	1	0

Tabla 3 Variables y operaciones del orden de la ruta

La figura 7 muestra el modelo introducido en el Solver de Calc.



Figura 7 El modelo en Solver de Calc

Si se hiciera un recorrido por los 40 clientes, la mejor ruta que rinde Solver, de 302.74 km, sería:

Fábrica, 1110, 429, 580, 457, 670, 1308, 1341, 1292, 1078, 1731, 1302, 1137, 1452, 1100, 408, 1167, 1713, 1636, 1133, 976, 1477, 1123, 343, 485, 1406, 1074, 1069, 894, 1363, 1392, 1319, 1321, 1138, 775, 722, 784, 924, 1533, 1580, 994, Fábrica.

Validación. Una de las desventajas de realizar cálculos en hojas electrónicas es su imprecisión numérica, que para fines cotidianos no tiene efectos importantes.

Sin embargo, los algoritmos exactos para resolver el TSP realizan una enorme cantidad de iteraciones que pueden alterar significativamente la precisión de los resultados, de modo que hay que tener cierta precaución sobre una finalización de Solver aparentemente óptima.

Por esta razón, una vez resuelto el modelo general, primero se verificó que la secuencia encontrada no contuviera subrutas y luego se agregó, repetidamente, una condición para reducir la distancia encontrada hasta que no hubiera solución factible.

Además se experimentaron optimizaciones de modelos TSP muy pequeños de hoja de cálculo de 5x5, 8x8 y 10x10 con datos de distancias tomadas de la matriz de 41x41, con el Solver de Excel, OpenSolver instalado en Excel y el Solver de Calc.

En todas las pruebas se obtuvieron los mismos resultados (distancia total mínima), por lo que se confía en que Calc efectivamente haya rendido la ruta óptima tanto en el caso general como en los tres cercanos a 20 clientes que realmente se realizaron y que sirvieron de contraste. En estos, la capacidad del SOLVER original de Excel fue insuficiente para resolverlos y OpenSolver tuvo que ser detenido al no rendir una solución antes de quince minutos.

Registro de recorridos actuales y soluciones del optimizador. Los recorridos de los jueves abarcan más o menos 20 clientes y se consiguió registrar la ruta seguida en tres ocasiones: dos de 19 clientes y una de 16. Estas rutas sirven para verificar que el optimizador rinde una secuencia más corta y por cuánto.

Sabiendo qué clientes fueron atendidos y el orden de visita, se obtuvieron tres duplicados del archivo de Calc del modelo general –el cual se debe conservar como plantilla y con una copia de resguardo– y, en cada uno, se eliminaron las filas y columnas correspondientes a clientes no visitados en sendos itinerarios.

El segundo cuadro (tabla 2) se copia en la misma hoja, en la que se codifica la ruta efectuada, calculándose la distancia total con los datos del primer cuadro (tabla 1).

Enseguida se introduce el modelo de optimización en Solver y se procede a resolverlo (a diferencia de Excel, en Calc no se guarda el modelo al cerrar el archivo, por lo que debe introducirse cada vez que se inicie una nueva sesión).

En la fórmula de las celdas de la matriz de orden la constante 41 se debe sustituir por el número de clientes más uno, y en Solver el correspondiente lado derecho, 40, hay que cambiarlo por la cantidad de clientes del itinerario.

Resultados

El gráfico 4 y la tabla 4 resumen los resultados del funcionamiento del modelo de distancias contrastado con los recorridos históricos recolectados, en tanto que la tabla 5 es la equivalente del modelo de tiempos.

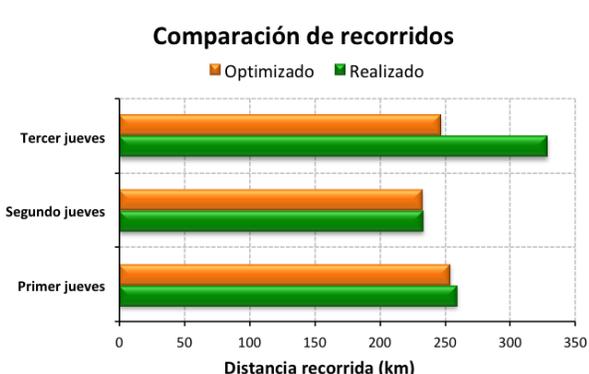


Gráfico 4 Resultados

La muestra es muy pequeña aún, al momento de escribir este reporte, para establecer con gran precisión el ahorro esperado en la distancia recorrida, pero hasta ahora está alrededor del 9%. Llama la atención que la longitud de la ruta del tercer jueves es mayor que la mínima de visita a todos los clientes, sin sugerir siquiera que es realizable esto último porque para ello se necesitaría estimar tanto el tiempo de traslado como el de entrega del producto. La tabla 5 es similar a la tabla 4, pero con mediciones de los tiempos de traslado en lugar de distancias. Dada la complejidad del análisis de esta variable, ya advertida en la subsección *Programación y resolución del modelo general*, los cálculos son realizados bajo el supuesto de que el tráfico fue el habitual. El ahorro esperado en tiempo es de aproximadamente 13%.

	Primer jueves		Segundo jueves		Tercer jueves	
	Realizado	Optimizado	Realizado	Optimizado	Realizado	Optimizado
1	Fábrica	Fábrica	Fábrica	Fábrica	Fábrica	Fábrica
2	429	429	580	580	994	429
3	457	457	457	457	924	670
4	670	670	1308	1308	429	1341
5	1308	1308	1731	1731	670	1292
6	1078	1100	1137	1137	1341	1078
7	1731	1078	1302	1302	408	1731
8	1302	1731	1100	1100	1100	1100
9	1100	1302	408	408	1292	408
10	1167	1713	1167	1167	1078	1713
11	1713	1167	894	894	1731	1074
12	1069	1069	1123	1123	485	485
13	894	1123	343	343	1713	1123
14	1123	485	1133	1636	894	894
15	485	894	1636	1133	1123	1363
16	1363	1363	1580	1580	1363	1392
17	1392	1392	994	994	1392	775
18	1321	1321	Fábrica	Fábrica	1074	784
19	722	722			775	924
	994	994			784	994
	Fábrica	Fábrica			Fábrica	Fábrica
Dist., km	259.20	254.00	233.44	232.44	328.70	246.95
Ahorro, en km			1.00		81.75	
En %			0.43%		27.87%	

Tabla 4 Resumen comparativo de distancias

Los modelos de Calc se corrieron en dos equipos de cómputo portátiles con procesadores Intel i5 e i7 y 8 y 16 Gb en RAM. El tiempo de corrida del modelo general varió entre dos y tres minutos y el de los casos de prueba fluctuó entre tres y quince segundos.

Con la debida capacitación de dos o tres empleados, con nivel mínimo de bachillerato, en el uso de la plantilla de Calc, en menos de veinte minutos puede generarse la mejor ruta de los jueves.

Este tiempo es prácticamente lo que se invertiría en identificar en la plantilla a los clientes a visitar y eliminar las filas y columnas no asociadas a ellos.

	Primer jueves		Segundo jueves		Tercer jueves	
	Realizado	Optimizado	Realizado	Optimizado	Realizado	Optimizado
	Fábrica	Fábrica	Fábrica	Fábrica	Fábrica	Fábrica
1	429	994	580	580	994	994
2	457	457	457	457	924	429
3	670	670	1308	1308	429	670
4	1308	429	1731	1731	670	1292
5	1078	1308	1137	1137	1341	1341
6	1731	1100	1302	1302	408	1078
7	1302	1731	1100	1100	1100	1731
8	1100	1302	408	1123	1292	1100
9	1167	1078	1167	343	1078	408
10	1713	1713	894	894	1731	1713
11	1069	1167	1123	1167	485	1123
12	894	1069	343	408	1713	485
13	1123	1123	1133	1636	894	1074
14	485	485	1636	1133	1123	894
15	1363	894	1580	994	1363	1363
16	1392	1363	994	1580	1392	1392
17	1321	1392	Fábrica	Fábrica	1074	924
18	722	1321			775	784
19	994	722			784	775
	Fábrica	Fábrica			Fábrica	Fábrica

Minu-tos	430	410	360	340	561	402
Ahorro, minutos	20.00		20.00		159.00	
En %	4.65%		5.56%		28.34%	

Tabla 5 Resumen comparativo de tiempos

Agradecimientos

Los autores agradecen a la fábrica Mr Hielo, y en particular a su jefe del área de administración y finanzas, el L.C. R. Eduardo León López, quien fue el impulsor del estudio y principal enlace con la planta, y al personal de la misma por el soporte y atenciones recibidas.

Conclusiones

Cuando se tiene una vasta experiencia en una determinada actividad que por sus características implique la resolución de un problema cada vez que se repite, una persona es capaz de encontrar buenas soluciones, incluso inmejorables si dicho problema es pequeño; pero a medida que aumenta el tamaño o complejidad del mismo, disminuye la probabilidad de alcanzar soluciones óptimas.

En el caso estudiado de Mr. Hielo, al conductor del camión repartidor se le da la lista de los clientes a los que debe entregar el producto y él decide el orden en que lo hará. De los tres viajes a cuya lista de clientes se tuvo acceso, el itinerario del repartidor no fue el de mínima distancia pero estuvo cerca: 259.2 km recorridos contra 254.0 del óptimo en el primero; 233.4 km contra 232.4 en el segundo. Sin embargo, el tercer jueves recorrió 328.7 km –¡mayor que la menor longitud del recorrido por los cuarenta clientes!– comparado con los 247.0 km del mínimo *tour*.

Dejar la decisión sobre el orden de entrega de las bolsas de hielo al repartidor no es un método consistente: a veces sí es buena pero también podría resultar muy mala. Una disminución media del orden del 10% en la longitud del *tour* con el uso del optimizador, abriría la posibilidad de aumentar en al menos un cliente la programación de las entregas al momento de planificar un viaje. Además, el optimizador permitirá programar y realizar el reparto de hielo aún en ausencia del chofer nominal.

Aunque se ha estimado en un máximo de veinte minutos la adecuación del itinerario en la plantilla y su optimización, con la práctica se volverá rutinario y más rápido.

Los resultados relacionados con el tiempo del recorrido y que evidencian un ahorro del 13% deben ser tomados con la debida reserva: i) son independientes del tiempo de demora en el recorrido debido a la entrega del producto; ii) el tiempo de viaje entre clientes es una variable aleatoria, en vez de constante como lo es la distancia; iii) sería necesario recurrir a otras técnicas de análisis, acompañantes de la optimización. A pesar de esto, tal reducción del tiempo es prometedora y anima a realizar las mediciones y análisis pertinentes de mejora.

La selección de OpenOffice Calc se debió a: i) es *software* libre y abierto; ii) no le resultaría complicado trabajar en él a un usuario familiarizado con las hojas de cálculo, que son de uso muy extendido; iii) su optimizador Solver, que ejecuta un algoritmo lineal exacto, tiene la capacidad suficiente para resolver el problema analizado; iv) el tiempo de resolución no excede quince segundos; y v) es multipataforma.

Recomendaciones

Para facilitar la implantación del nuevo método de definición de la ruta de distribución de los jueves, sería conveniente:

- Instalar el modelo de hoja de cálculo en un equipo de cómputo de escritorio de la empresa, debidamente respaldado en otro.
- Capacitar a por lo menos dos empleados de Mr. Hielo en su uso actual y en el procedimiento a seguir en caso de aumentar la cartera de clientes de la zona.
- Probar las rutas generadas por el modelo durante algunos viajes más y, si su eficiencia es convincente, extender su uso a otras zonas de distribución del hielo.
- Cuando económicamente se justifique, mejorar la utilidad del modelo mediante la incorporación de tecnologías de la información en tiempo real.

Referencias

Clay Mathematics Institute. (18 de Agosto de 2007). *Millennium Problems*. Recuperado el 7 de Agosto de 2018, de CMI (Clay Mathematics Institute): <http://www.claymath.org/millennium-problems>

Chondl, M. (Septiembre de 2012). An Integer Programming Quiz Picture Round. *Transactions on Education* , 65-66.

Chondl, M. (Mayo de 2016). TSP at the Movies: Yondu's Dart Problem. *Transactions on Education* , 110-111.

Cook, S. (8 de Abril de 2006). *P vs NP problem*. Obtenido de CMI (Clay Mathematics Institute): <http://www.claymath.org/millennium-problems/p-vs-np-problem>

Cook, W. J. (2012). *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton, NJ, USA: Princeton University Press.

Craig A., T. (2002). Tutorial on Computational Complexity. *Interfaces* , 30-61.

Bosch, R. (2010). *Mathematical Imagery*. Recuperado el 7 de Agosto de 2018, de AMS American Mathematical Society: <http://www.ams.org/mathimagery/displayimage.php?pid=274>

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America* , 393-410.

Hillier, F., & Lieberman, G. (2010). *Introducción a la Investigación de Operaciones* (9ª edición ed.). México: McGraw-Hill.

Holland, C., Levis, J., Nuggehalli, R., Santilli, B., & Winters, J. (Enero-Febrero de 2017). UPS Optimizes Delivery Routes. *Interfaces* , 8-23.

Orman, A., & Williams H., P. (2005). *A survey of different integer programming formulations of the travelling salesman problem*. London School of Economics and Political Science, Department of Operational Research. London, UK: Operational Research working papers, LSEOR 04.67.

Taha, H. (2012). *Investigación de Operaciones* (9ª edición ed.). México: Pearson.

Wikipedia. (2 de Agosto de 2018). *Seven Bridges of Königsberg*. Recuperado el 2 de Agosto de 2018, de Wikipedia The Free Encyclopedia: https://en.wikipedia.org/wiki/Seven_Bridges_of_Königsberg