

Investigación de Operaciones en acción:

Aplicación del TSP en problemas de manufactura y logística

José Luis González Velarde*
Roger Z. Ríos Mercado**

Resumen

En este artículo se describe uno de los problemas más famosos y difíciles en la teoría de optimización. Se hace una breve reseña de métodos que se han propuesto para su solución, se presentan también algunas aplicaciones prácticas, en particular el Problema del Agente Viajero, y se esboza brevemente el significado de la expresión Complejidad Computacional.

Palabras clave: Investigación de Operaciones, Optimización, Algoritmo, Heurística.

1. INTRODUCCIÓN

La ciencia de la toma de decisiones, mejor conocida como Investigación de Operaciones (IO), nació hace ya más de cincuenta años cuando George Dantzig inventó el método Simplex para resolver problemas de optimización lineal, es decir, problemas cuyas variables de decisión son continuas y relacionadas de manera lineal. Aun cuando en sus orígenes, esta naciente área de la ciencia fue motivada por aplicaciones de carácter militar, la IO fue alcanzando un alto grado de interés entre



investigadores y profesionistas en los campos de ingeniería, matemáticas aplicadas y administración, quienes motivados por los diversos y complejos problemas de toma de decisiones que surgían en varias áreas del quehacer científico e industrial, comenzaron a estudiar y desarrollar metodologías de solución para problemas de diferentes características. Fue así como nacieron posteriormente las ramas de optimización no lineal (relación no lineal entre las variables de decisión), optimización discreta (variables enteras) y optimización entera mixta (en variables continuas y discretas), por mencionar algunas.

Aplicaciones de IO se encuentran en prácticamente todos los niveles y en todo tipo de industrias. Es evidente que las corporaciones aspiran a tomar decisiones que les reditúen beneficios económicos, y normalmente, estas decisiones se encuentran restringidas de forma muy compleja. Estos atributos son únicos de modelos de IO. En las últimas décadas el impacto de IO en la industria ha sido impresionante, convirtiéndose en ganancias (o ahorros) con frecuencia multimillonaria en los diversos ramos industriales.

El presente es el primero de una serie de artículos que pretenden introducir al lector con problemas y metodologías de IO (clásicas y recientes) y cómo éstas se usan para resolver problemas reales que surgen en los diversos campos de la ciencia: ingeniería química, ingeniería civil, ingeniería eléctrica, administración, economía, ciencias computacionales, estadística y matemáticas aplicadas entre otras. Así mismo se pretende ilustrar la importancia de saber evaluar las ventajas y desventajas que surgen entre la obtención de soluciones de alta calidad contra los recursos

* Profesor visitante, Universidad de Colorado, Escuela de Graduados en Negocios y Administración.

** Investigador Asociado, Texas A&M University, Departamento de Ingeniería Industrial.

empleados para obtenerla (tiempo de cómputo, requerimientos de memoria).

En este artículo, en particular, tratamos un problema clásico de IO como lo es el Problema del Agente Viajero (TSP, por sus siglas en inglés: Traveling Salesperson Problem) y su aplicación para resolver el problema de programación de tareas que se presenta en la manufactura, y el del ruteo de vehículos en el ramo de la logística.

En la Sección 2 introducimos al lector con el TSP, describiendo la razón por la cual este problema, tan sencillo de formular, es muy difícil de resolver y mencionamos brevemente las técnicas más eficientes para resolverlo. En la Sección 3, planteamos algunas de las aplicaciones más importantes del TSP en varios tipos de industrias. Concluimos en la Sección 4 con comentarios finales.

2. ¿QUÉ ES EL TSP?

El TSP,¹ uno de los problemas clásicos de optimización, se formula de la siguiente manera. Un agente viajero, partiendo de su ciudad de origen, debe visitar exactamente una vez cada ciudad de un conjunto de ellas (previamente especificado) y retornar al punto de partida. Un recorrido con estas características, es llamado dentro de este contexto un *tour*. El problema consiste en encontrar el *tour* para el cual la distancia total recorrida sea mínima. Se asume que se conoce, para cada par de ciudades, la distancia entre ellas. La Figura 1 ilustra un *tour* en una instancia de ocho ciudades.

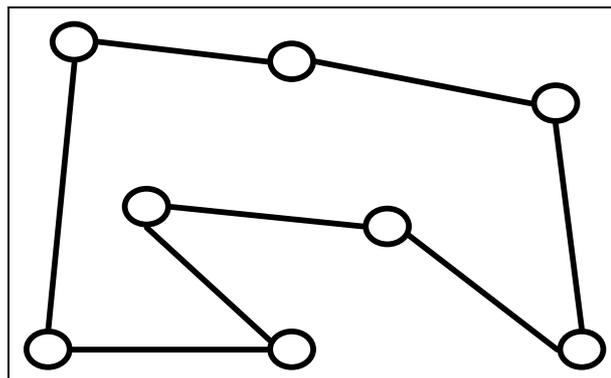


Fig. 1. Un tour en un TSP de ocho ciudades.

El problema en sí es fácil de formular. Sin embargo, al igual que muchos otros que se presentan en el campo de optimización, es sumamente difícil de resolver (por resolver, nos referimos a encontrar la solución óptima al problema y probar desde luego que ésta es efectivamente la mejor solución posible). El establecer cuándo un problema es “fácil” o “difícil” (la cual es una de las áreas más importantes en los campos de optimización y computación) está íntimamente ligado al tiempo de solución del problema. Sin entrar en detalles técnicos, decimos que un problema es “fácil” de resolver cuando es posible encontrar un algoritmo (método de solución) cuyo tiempo de ejecución en una computadora crece de forma “razonable” o moderada (o polinomial) con el tamaño del problema. Por el contrario, si no existe tal algoritmo decimos que el problema es “difícil” de resolver. Esto no implica que el problema no pueda resolverse, sino que cada algoritmo existente para la solución del problema tiene un tiempo de ejecución que crece explosivamente (o exponencialmente) con el tamaño del problema. La consecuencia directa de un algoritmo que tiene una función de tiempo exponencial es que a medida que aumenta el tamaño del problema, el tiempo requerido para la solución aumenta de forma exponencial, lo cual limita bastante el tamaño de

problemas que pueden resolverse en las computadoras modernas. Técnicamente hablando, determinar si un problema es fácil o difícil se denomina establecer la complejidad computacional del problema, y esto es todo un arte, especialmente para demostrar que un problema es de los difíciles. Para un estudio más a fondo sobre complejidad computacional, recomendamos la obra de Garey y Johnson.¹

Veamos un ejemplo. Supongamos que tenemos una instancia del TSP con n ciudades. Una forma (poco inteligente) de resolverlo es por enumeración exhaustiva. Es decir, formamos todas las posibles combinaciones de tours (en este caso $(n-1)!$, donde $n! = n(n-1)(n-2)\dots(2)(1)$) y calculamos la distancia total para cada tour, eligiendo aquel que tenga la mínima distancia total. En este caso el problema ha quedado totalmente resuelto porque estamos exhibiendo todos los tours posibles. El tiempo de ejecución de este algoritmo es a grosso modo $f(n)=(n)!$

Esta forma, como puede verse, deja de ser viable una vez que consideramos conjuntos de ciudades mayores. En el caso $n=5$, por ejemplo, tenemos que calcular $4!=24$ tours lo cual puede hacerse en fracción de segundos en cualquier computadora. Al considerar un conjunto con $n=50$ ciudades, el número posible de tours es $49!$, el cual es un número tan gigantesco que no alcanzaría a resolverse en varios meses ni en las computadoras más potentes de hoy en día. Hay que notar que la función factorial $f(n)=n!$ es una función que crece exponencialmente a medida que crece el valor de n . Claro, esto no prueba que el TSP es difícil, ya que muy bien pudiera existir otro algoritmo que lo resolviera cuyo tiempo de ejecución fuera polinomial. En este caso, sin embargo, ya se ha demostrado que tal algoritmo polinomial no existe y que el TSP pertenece a esa clase de problemas difíciles. La Figura 2 (tomada de¹) ilustra las diferencias de crecimiento de diferentes funciones de tiempo (columnas).

Las cifras que se muestran son tiempo de procesamiento en computadora que procesa 1 millón

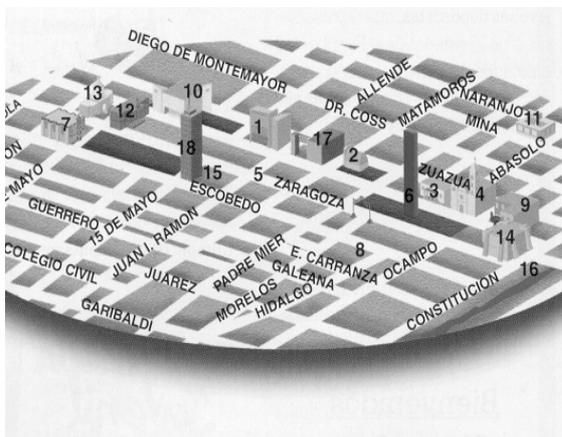
| Tamaño n | $f(n)=n$ | $f(n)=n^2$ | $f(n)=n^3$ | $f(n)=n^5$ | $f(n)=2^n$ | $f(n)=3^n$ |
|------------|---------------|--------------|-------------|----------------|-----------------|--------------------------------|
| 10 | .00001 seg | .0001 seg | .001 seg | .1 seg | .001 seg | .059 seg |
| 20 | .00002 seg | .0004 seg | .008 seg | 3.2 seg | 1.0 seg | 58 minutos |
| 30 | .00003 seg | .0009 seg | .027 seg | 24.3 seg | 17.9 minutos | 6.5 años |
| 40 | .00004 seg | .0016 seg | .064 seg | 1.7 minutos | 12.7 días | 3855 siglos |
| 50 | .00005 seg | .0025 seg | .125 seg | 5.2 minutos | 35.7 años | 2×10^8 siglos |
| 60 | .00006 seg | .0036 seg | .216 seg | 13 minutos | 366 siglos | 1.3×10^{13} siglos |

Fig. 2. Comparación de varias funciones polinomiales y exponenciales.

de operaciones de punto flotante por segundo. Nótese el crecimiento explosivo de las funciones exponenciales (últimas dos columnas).

La simplicidad de su formulación aunado a la dificultad en resolverse fueron los factores que atrajeron a investigadores en las áreas de matemáticas discretas a estudiar el problema y desarrollar técnicas especiales. En 1990, la instancia del TSP de mayor tamaño que había sido resuelto era una de 318 ciudades.² Hoy en día existen métodos basados en técnicas de ramificación y corte/acotamiento, las cuales explotan muy efectivamente la estructura matemática del problema, que han sido muy exitosas. En 1998, se reportó la instancia más grande que se ha resuelto de 13,509 ciudades,³ lo cual evidencia el tremendo progreso logrado durante la década de los noventa. Algunas de estas técnicas serán tratadas con más detalle en artículos posteriores.

Desde luego que hay también razones prácticas que hacen importante el TSP. Muchos problemas reales pueden formularse como instancias del TSP, como se verá en la siguiente sección.



3. APLICACIONES DEL TSP

Describiremos ahora algunos problemas que se presentan de forma natural en algunas empresas. El primero tiene que ver con la programación de tareas en una máquina. Muchas veces en algún taller de manufactura, se cuenta con una sola máquina en la cual se pueden procesar diferentes tareas, una a la vez. Ahora bien, para procesar cada una de estas tareas, la máquina requiere de cierta configuración característica de la tarea, pueden ser: número y tamaño de diferentes dados, colocación de cuchillas a cierta distancia unas de otras, colorantes para alguna fibra, etc. De manera que una vez que una tarea ha sido terminada, es necesario preparar la máquina para procesar una nueva tarea, aquí será necesario invertir un cierto tiempo, y este tiempo dependerá de la tarea recién procesada y de la próxima. Si las características de una tarea son similares a las de otra, es plausible pensar que el tiempo que se requiere para pasar de una configuración a otra será pequeño, en comparación del tiempo requerido para pasar de una tarea a otra con características muy diferentes.

Desgraciadamente durante las labores de preparación de la máquina, ninguna de las tareas se puede ejecutar, así que este tiempo es tiempo perdido, y se está desaprovechando la capacidad de la máquina, esto representa un costo de oportunidad para la empresa. Es importante entonces encontrar el orden en el cual se deben de procesar estas tareas con el fin de reducir al mínimo todo este tiempo perdido.

Aún cuando este problema parezca no tener ninguna relación con el TSP, se puede formular de la misma manera. Cada tarea puede ser vista como una de las ciudades a visitar, y el tiempo necesario para cambiar la configuración de la máquina corresponde a la distancia que hay entre una ciudad y otra. Encontrar la manera de ordenar las tareas para minimizar el tiempo total de preparación es equivalente a diseñar la ruta, esto es, el orden en el cual se deben de visitar las

ciudades para minimizar la distancia total recorrida. Esto nos da una idea de lo crucial que resulta tener buenas soluciones para el TSP en un ambiente de manufactura.

Un segundo ejemplo lo podemos encontrar dentro de la logística de distribución de mercancía a los clientes. Generalmente, algunas empresas que distribuyen bienes perecederos necesitan hacerlo en un tiempo corto, un esquema muy común es que la empresa disponga de un almacén central, en el cual se concentran los bienes a distribuir, y una flotilla de unidades de transporte se encarga de visitar a los clientes para hacer entrega de la mercancía.

Analicemos los componentes de este problema, en primer lugar tenemos que las unidades de servicio son limitadas, la forma en la que se podría efectuar la entrega de mercancías en el menor tiempo posible, sería enviar una unidad a cada uno de los clientes. Pero, lo más realista sería pensar que no se tienen tantas unidades como clientes, ya que esto resultaría sumamente oneroso. Si la empresa dispone de una sola unidad el costo fijo se reduce bastante, y el problema de determinar la ruta que debe de seguir el vehículo para entregar en el menor tiempo toda la mercancía es ni más menos que el TSP. Pero aquí hay dos problemas en los que tenemos que pensar: en primer lugar, puede ser que el tiempo mínimo (si es que se puede determinar) resulte demasiado largo, p. ej. si se trata de entrega de leche, esta debe de estar entregada por la mañana, que es cuando los clientes la requieren, y con una sola unidad de entrega, podría darse el caso que los últimos clientes la fueran recibiendo por la tarde. Por otro lado, las unidades tienen una cierta capacidad de almacenamiento, y puede ser que

se necesiten varias para poder cargar con toda la mercancía que debe de ser entregada.

Así pues vemos que este problema contiene dentro de sí muchos más. Primero: determinar cuál es el tamaño ideal de la flota de vehículos. Segundo: determinar cuáles son los clientes que deben de ser asignados a cada unidad para hacer la entrega. Y finalmente: cuál es la ruta que debe de seguir cada una con el fin de terminar con el reparto en el menor tiempo posible (TSP). Para complicar más las cosas estos problemas no son independientes, sino que la solución de uno determina la de otro. Este problema se conoce como el problema de ruteo de vehículos (VRP: Vehicle Routing Problem). Muchas aplicaciones más pueden encontrarse en el libro de Lawler *et al.*² La Figura 3 ilustra un ejemplo de ruteo.

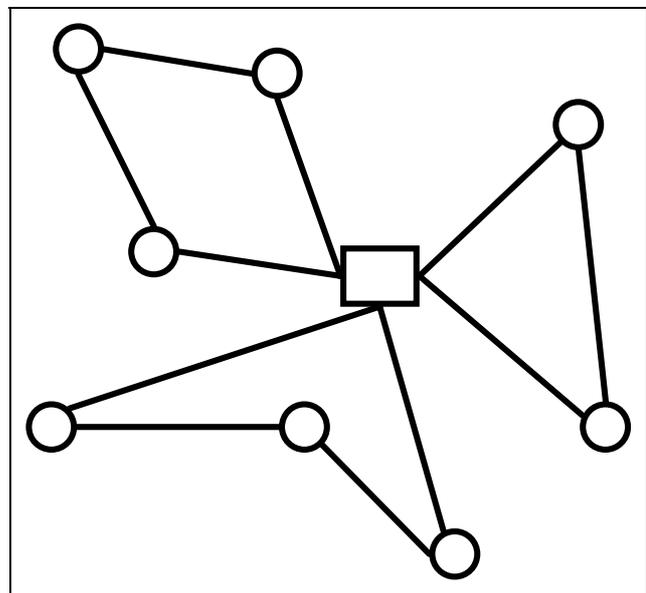


Fig. 3. Ejemplo de un ruteo factible en un VRP (una central de abasto, ocho clientes y tres unidades de servicio).

4. CONCLUSIÓN

En este artículo hemos presentado al TSP, uno de los problemas clásicos de IO, así como su aplicación en problemas logísticos y de manufactura. Hemos también señalado cómo es que la búsqueda de procedimientos para resolver efectivamente problemas como éste, conlleva a un significativo avance en la operación óptima de las empresas o industrias donde se presentan este tipo de problemas. Con respecto a esto, es importante señalar lo vital que resulta la labor de equipo para enfrentar y resolver exitosamente problemas de toma de decisiones. Se requiere de personal que tenga el conocimiento y entendimiento del fenómeno/problema que se pretende resolver, así como también personal experto en IO que tenga la preparación técnica suficiente para modelar, y proponer/desarrollar técnicas adecuadas de solución explotando la estructura matemática del problema. Una comunión exitosa se traduce en un impacto significativo de carácter económico para la empresa.

Lo más alentador es que, a pesar de los marcados avances en cuestión de teoría, metodologías y aplicaciones, aún hay una amplia área de oportunidad para efectuar avances todavía mayores en esta importante área del conocimiento. Otros modelos, metodologías y aplicaciones serán tratados en artículos posteriores.

REFERENCIAS

1. M. R. Garey y D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
2. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan y D. B. Shmoys, editors. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley, Chichester, 1985.
3. *Optima*, 58, Mathematical Programming Society Newsletter, Junio 1998.